

Qualcomm
thinkabit lab

Troubleshooting Guide

Table of Contents

Safety Considerations	5
Connecting wires to breadboard	6
Connect LED	7-8
Connect Servo	9
Connecting to Arduino	10
Corrupt/Crashing Arduino IDE	11
Write Code: Include Library	12
Write Code: Declare Variable	12
Write Code: void setup	13
Write Code: void loop	13
Upload Code	14
AppInventor	15
Connect to Bluetooth	16
Testing App on Device	17
Diagram of App Inventor/Arduino/BT Set-up	18

Safety Considerations

Tell students not to touch the exposed ends of the wires together while connected to a powered Arduino board. A short circuit is created when the grounded wire is touched to the electric current wire while connected to a powered Arduino board. A short circuit can potentially cause the wires and/or the Arduino board to get sufficiently hot to burn the skin. In addition, in the event of a short circuit the Arduino board can potentially catch fire.

Connecting wires to breadboard

1. I don't have a red/black wire
 - While the actual color of the wire does not matter, color coding wires makes it easier to use for instruction.
2. There's a wire stuck in Vin/GND
 - An instructor or adult can use pliers to remove the wire. Allow the board 1-2 minutes after unplugging before removing the wire
 - Or, replace the Arduino
 - Or, use a different GND
3. Wire doesn't fit in the breadboard or Arduino
 - Use 22-gauge wire
 - Use a different hole in the same row on breadboard
4. Wire doesn't have enough exposed metal to connect to the Arduino or breadboard
 - An instructor or adult can use the wire cutters to strip the plastic coating off the end of the wire

Connect LED

1. Don't have white wire
 - While the actual color of the wire does not matter, color coding wires makes it easier to use for instruction.
2. There is a wire stuck in Vin/GND
 - An instructor or adult can use pliers to remove the wire. Allow the board 1-2 minutes after unplugging before removing the wire
 - Or, replace the Arduino
 - Or, use a different GND
3. Wire doesn't have enough exposed metal to connect to the Arduino or breadboard
 - An instructor or adult can use the wire cutters to strip the plastic coating off the end of the wire
4. Wire/Resistor/LED doesn't fit into the breadboard
 - Use 22-gauge wire
 - Try a different hole in the same row of the breadboard
5. Resistor legs are bent
 - The legs of the resistor can be bent or reshaped, as needed
 - Replace the resistor if it breaks. It is normal for the resistor to break after repeated reshaping of the legs.
6. LED did not light up
 - Follow the student's signal wire for the LED (usually pin 13) and power it through Vin, 5V, or 3.3 → this will show whether the circuit is working
 - No Bluetooth:
 - The LED is programmed to be controlled by the information sent over the Arduino → check that the LED's long and short leg as well as the necessary wires are placed in the correct place
 - If nothing lights up, change the LED
 - With Bluetooth:
 - The LED is programmed to be controlled by the information sent over the Bluetooth, so the LED should not turn on until a signal from app inventor is sent.
 - Ensure characters sent from Arduino and Bluetooth are in 'single quotes', and not "double quotes".

7. LED is not blinking during blinking activity
 - Make sure white wire (or designated wire) is plugged into a digital pin (usually pin 13) and double check the delay the student programmed in the code.
 - Student might have:
 - Plugged wire into a different digital pin but coded for pin 13
 - Plugged wire into pin 13 but coded for a different pin (watch out for white wire and coding for a PWM pin)
8. LED legs are bent
 - Bend and reshape the legs of the LED, as necessary.
 - It is normal for LED legs to break after repeated reshaping of the legs. Replace the LED.
9. Can't find the long/short leg of LED
 - When the LED legs are reshaped many times, it could be difficult to figure out which leg is short or long.
 - Placing the LED in backwards won't have a negative effect on the LED, it just won't turn on, so you can try it one way and if it doesn't work, flip it around.

Connect Servo

1. The wires keep falling out of the Servo connector
 - Experiment holding the wires with your fingers to check the connection and use electrical tape for the final project.
 - If bent, straighten the wire with your fingers so the wire goes in further and the red and black wire do not touch.
 - Try a different servo.
 - Check that solid 22-gauge wire is used.
2. Don't have red/black/white wire
 - While the actual color of the wire does not matter, color coding wires makes it easier to use for instruction.
3. There's a wire stuck in pin 9
 - An instructor or adult can use pliers to remove the wire. Allow the board 1-2 minutes after unplugging before removing the wire
 - Or, replace the Arduino
 - Or use another PWM pin and modify your code to connect the Servo to the new pin.
4. Wire doesn't have enough exposed metal to connect to the Arduino or breadboard
 - An instructor or adult can use the wire cutters to strip the plastic coating off the end of the wire
5. Wire doesn't fit into the breadboard
 - Use 22-gauge wire
 - Use a different hole in the same row of the breadboard.

Connect to Arduino

1. The sketch has code already
 - To get a new Arduino sketch window, go to "File" and click "New".
2. My COM# is different than the one in the video
 - That is normal. Each time you connect a new USB device, a new com (communication port) number is assigned to it. Yours will likely be different than the video and even different from your classmates.
3. No COM# (Arduino Uno)/Grayed out "Port"
 - Make sure the USB is plugged all the way into both the Arduino board and the computer USB port.
 - Click out of the menu bar and try again (sometimes it takes a while for the computer to recognize a new device).
 - Unplug and re-plug the USB cable.
 - Go to Device Manager
 - Expand "Ports (Com and LPT)"
 - If COM# is greyed out, right-click and "Uninstall"
 - If Ports (Com and LPT) is not visible:
 - Go to View in menu bar
 - Click 'Show Hidden Devices'
 - Restart the Arduino software.
 - Use a new Arduino Uno board.
 - Use a different USB cable.
 - Use a different computer.
4. Green light is not on
 - Change power cord
 - Change Arduino
5. Orange LED light (next to pin13) is not on/blinking
 - If NOT using LED blink library, this light should be blinking because it indicates pin13's ability to function.
 - Use a different digital (but not PWM) pin
 - Change Arduino

Corrupt/Crashing Arduino IDE

This can manifest in many ways: greyed out "Port" selection on Arduino; Arduino begins to open but closes itself down and crashes; microcontroller is unresponsive to software; etc.

1. Check if Windows has had a recent (ex. overnight) automatic update
 - At times, Windows updates are not compatible with Arduino software since it is an open-source coding platform.
2. COMPLETELY uninstall Arduino from computer (warning: any code you or students have worked on will be lost. Encourage them to write code in notebook first in case this happens)
 - Unplug Arduino from computer to prevent corrupting the hardware
 - Control Panel → Programs → Uninstall Programs → Right click "Arduino" to uninstall
 - Recycle Bin → Empty recycle bin (to remove any Arduino files)
 - Double-check C: Drive → Program Files (x86) [this is where Arduino is usually housed as well]
 - C: Drive → Users → Your username → AppData → Local → Delete "Arduino15"
 - This folder will sneak its way back into this location, so make sure to look back again after reinstalling and delete
 - Restart computer
 - Reinstall Arduino from <https://www.arduino.cc/en/Main/Software> and use the "Windows Installer" link
 - Test
 - If issues persist, consider changing computer, microcontroller, and power cord.

Write Code: Include Library

1. Can't find 'Servo' or "SoftwareSerial" in Sketch>Include Library
 - It's there; keep looking!
2. Can't find "#include <Servo.h>" or "#include <SoftwareSerial.h>"
 - Check the very top of the sketch code

Write Code: Declare Variables

1. Colors don't change like they do in the video
 - The words "Servo", "SoftwareSerial" and "int" should all change colors as they are recognized functions in Arduino. If they did not change colors as shown in the video, check the following:
 - Check the spelling. Are the words spelled correctly with the proper capitalization?
 - Check the spacing. Is proper spacing used? Watch the video and read the instructions carefully.
 - If it changed color to a different color, the Arduino software may have been updated. If these words are colored, the code should still work properly.

Write Code: Void Setup

Colors don't change like they do in the video. The words "begin", "pinMode", "OUTPUT" and "attach" should all change colors as they are recognized functions in Arduino. If the colors do not change as shown in the video, check the following:

- Check the spelling of the words and for proper capitalization.
- Check the spacing, watch the video and read the instructions carefully.
- If the color changed, but to a different color than the video, the Arduino software may have been updated. If these words are colored, the code should still work properly.

Write Code: Void Loop

1. There are extra closed brackets at the end of the code
 - Double check that you have all the brackets needed to complete the loop function and then delete any extra brackets.
2. Colors don't change like they do in the video

Several words should change colors as they are recognized functions in Arduino. If they did not change colors as shown in the video, check the following:

- Check the spelling of the words and for proper capitalization.
- Check the spacing, watch the video and read the instructions carefully.
- If the color is changed, but to a different color than the video, the Arduino software may have been updated. If these words are colored, the code should still work properly.

Uploading Code

1. "Done compiling" is not the same as "done uploading"
 - "Done compiling" is a feature used to test the code's ability to compile before sending to the board. Once compiling is finished, make sure that you are clicking on the "upload" button.
2. "Expected declaration before '}' token"
 - You have too many brackets in the code. Make sure every closed bracket corresponds to an open bracket.
3. "Expected '}' at end of input"
 - You are missing a bracket in the code. Make sure every open bracket corresponds to a closed bracket.
4. "Expected ')' before ';' token"
 - Notice that the Arduino software highlights the area with the problem.
 - You might be missing a parenthesis. Make sure every parenthesis has its match.
5. "Expected ';' before '____'"
 - Notice that the Arduino software highlights the area with the problem.
 - You might be missing a ';' at the end of a line of code. Make sure each line has a ';' after it.
6. "Error compiling"
 - Click 'upload' again.
 - Double check the code. Make sure everything is included from the code demonstrated in the earlier steps.
7. "Problem uploading to board"
 - Make sure that the Arduino board is connected to the computer with the USB cable.
 - Make sure that the Arduino is connected to the right COM port, then upload again.
 - To check, go to Tools > Port and see if "COM# (Arduino/Genuino Uno)" is next to the word "Port".
 - If not, select the option mentioned above.
 - If no COM# shows "(Arduino/Genuino Uno)" next to it, make sure that the USB cable is connected to both the Arduino board and the USB port of the computer.
 - Unplug and re-plug the USB cable and upload again.
 - Disconnect all wires from the Arduino board and upload again.
 - If using Bluetooth, disconnect power from Bluetooth as well since code will not upload while pins 0/1 are powered.

AppInventor

1. Use Google Chrome as the default browser.
2. Don't have a Gmail account?
 - Go to gmail.com and sign up! You'll need this account to access App Inventor. The apps created will be stored under that email address.
3. Don't have an Android device, rather I have an Apple (iOS-based) device, can I use it with App Inventor?
 - No, App Inventor is currently only compatible with Android-based devices. Although many apps are available on both iOS and Android, the coding process for these 2 platforms is quite different.
4. Missing any of the required materials?
 - You will not be able to complete this activity without the required materials. Please acquire all materials before beginning this activity.
5. Don't have jumper wires, rather, only have red, black and white wires from Thinkabit Lab starter kit?
 - Purchase jumper wires from Amazon, Sparkfun, or another retailer
 - Use the red, black, and white wires with the breadboard to make your connections (look through the troubleshooting section for tips on making these connections with the Bluetooth module).
6. Can't save my app with the name I want
 - The app name cannot contain any spaces. Use an underscore (_) to create a space or use the name "BluetoothApp" with no spaces.
 - The app name should be unique and cannot be a name that has previously been used by your Gmail account. Use a different name or add a number at the end.
 - The app name should only start with a letter, it cannot start with a number.
7. Don't see the component I want in the Palette
 - In the top left corner, there is a textbox where you can search for components.
 - If issue persists, watch the video and pause it as needed to see where to find each component.
8. Don't see where in the Properties section to change the text, width, background color, etc.
 - Be sure that you are modifying the properties of the correct component, click on the component you want to edit (in the Components section or on the viewer) to modify the component's properties.
 - Scroll down to find it. It is there!
9. Having trouble finding some of the blocks needed to build this code
 - Click a blank area on the screen and begin typing the words of the block you want. It should auto populate options and then you can select from there.
 - If issue persists, watch the video and pause as needed to find the location of each block.

Connect Bluetooth Module

Pins 0 and 1 on the Arduino (RX and TX, respectively) are for communication between the Arduino and other devices. This includes the Bluetooth device connected, as well as the computer connected by the USB cable. When another device is connected to pins 0 and 1, the ability to communicate with the computer is lost. Therefore, any changes made to the Arduino code is not uploaded and the Arduino is not able to communicate with the computer while the device is connected to pins 0 and 1. If you want to upload new code, disconnect Bluetooth module from power by disconnect the wires from 0 and 1 to ensure Tx/Rx don't get mixed up. Then, upload the code and reconnect the wires only after the code is done uploading.

Alternatively, you could modify the code to connect the Bluetooth module via different pins. You must use PWM pins, for example 10 and 11 which is an easy change from 0 and 1. To make this change in the code, edit the line that says "SoftwareSerial BTConnect(0, 1);" to "SoftwareSerial BTConnect(10, 11);". Please note that 0 and 1 were selected because for their ease of use. If you want to create 2-way communication between the Android device and the Arduino, it is recommended to use pins 0 and 1 for the Bluetooth connection.

1. Don't have jumper wires

- Place the Bluetooth module in the breadboard at the opposite end from the colored rows, being sure to place each connection in a different row.
- Get 4 more wires, 1 red, 1 black, and 2 white.
 - Connect the red wire from the row of the breadboard where VCC is connected to 5V on the Arduino.
 - Connect the black wire from the row of the breadboard where GND is connected to GND on the Arduino.
 - Connect one white wire from the row of the breadboard where TXD is connected to Pin 0 (RX) on the Arduino.
 - Connect the other white wire from the row of the breadboard where RXD is connected to Pin 1 (TX) on the Arduino.

2. There's a wire stuck in Vin/GND/Pin 0/1

- An instructor or adult can use a box cutter or pliers to remove the wire
- Or, replace the Arduino

Testing App On Device (Tablet)

1. Mobile device will not connect to App Inventor
 - Make sure Google Chrome browser is being used on the computer when working with App Inventor as it is the preferred browser.
 - Make sure that the device is connected to Wi-Fi to use the AI2 Companion app.
 - Make sure that the six-character code is entered correctly and “connect with code” has been pressed or that the QR code has been scanned.
 - Make sure both the computer and the android device are on the same WiFi network.
 - Sometimes it takes time for the mobile device to connect. Wait few minutes before doing any corrective measures.
 - The program might be frozen. Close the program completely and restart again. The connection on the computer might need to be reset as well by going to Connect and clicking “Reset Connection”.
2. Mobile device won't connect to Bluetooth module
 - Make sure that the Arduino is connected to power and the Bluetooth module is properly connected to the Arduino.
 - Make sure that you are attempting to connect to the correct Bluetooth module.
 - Press the reset button on the Arduino.
 - Press the reset button on the Bluetooth module.
 - Disconnect the Bluetooth module from the jumper wires and reconnect again.
3. Pressing the buttons and nothing is happening
 - Check for errors systematically, like an engineer.
 - Use Vin, 5V, or 3.3V to test if you can power LED. This checks if the problem is with the circuit.
 - Make sure the Android device is connected to the Bluetooth module.
 - Check the App Inventor blocks to make sure that each button is programmed to send information over Bluetooth.
 - Check the Arduino code to make sure that each letter received over Bluetooth triggers an action on the LED or Servo.
 - Check that all wires have been properly connected, students often flip-flop the placement of the wires for TXD and RXD.
 - Re-check the Arduino code to make sure that all lines of code are properly written, repeat earlier steps as necessary.
 - Press the reset button on the Arduino.
 - Upload the code to the Arduino again (remember to disconnect the wires from pins 0 and 1 AND the Bluetooth module before uploading and only to reconnect the wires once the code is done uploading).

Diagram of Arduino/App Inventor/Bluetooth Set-up

