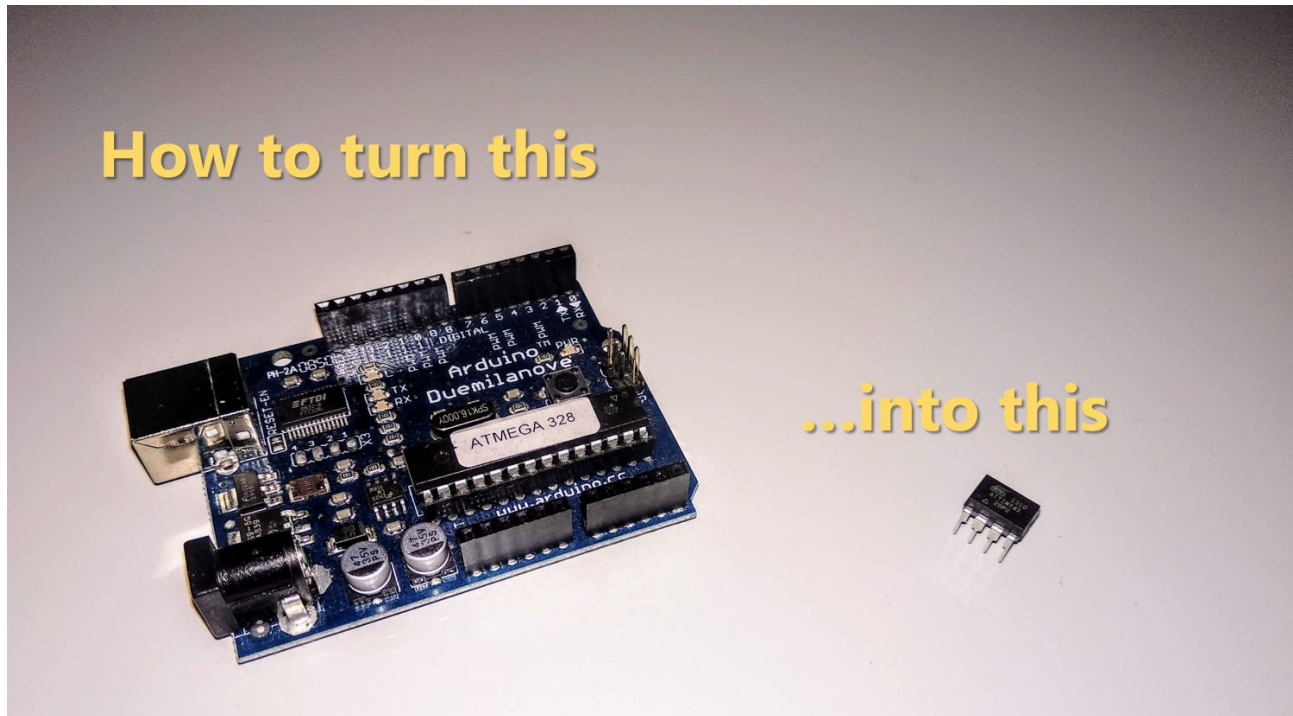


Shrink your microcontroller project using Arduino ISP

Course material, v1.2



| | |
|--|----------|
| Step 1: Run demo sketch on the Arduino UNO | 1 |
| Demo circuit (no ISP) | 1 |
| Step 2: Uploading the ISP software to the Arduino UNO | 2 |
| Step 3: Installing ATtinyCore | 3 |
| Step 4: Wiring Arduino UNO as ISP for ATtiny45 | 7 |
| ATtiny45 pinout | 7 |
| Arduino as ISP with demo circuit | 8 |
| Step 5: Final stand-alone demo circuit | 9 |

Step 1: Run demo sketch on the Arduino UNO

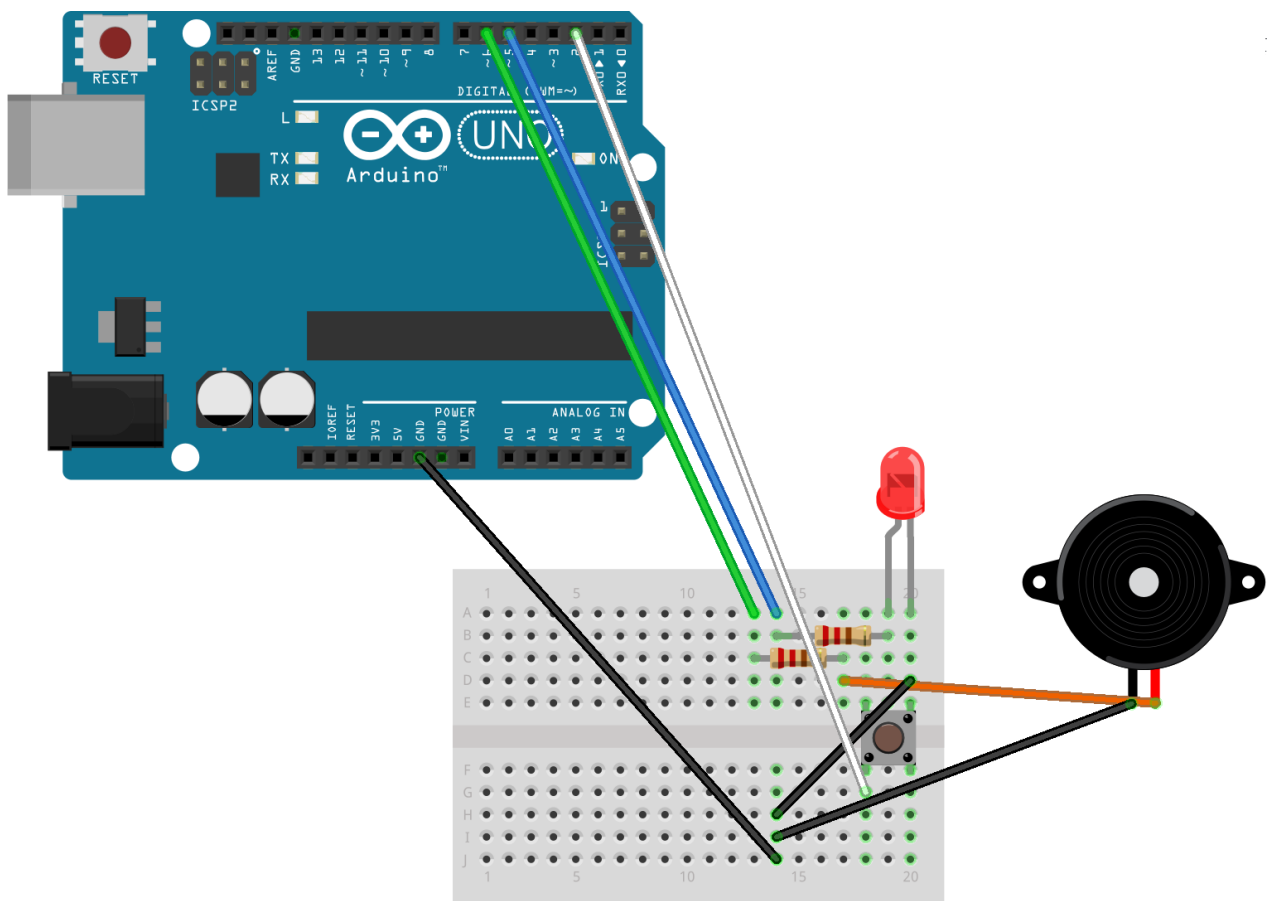
This is the prototyping phase, in which we develop our project using the Arduino UNO as our (only) controller in the circuit.

In this workshop we will eventually eliminate the Arduino entirely from our circuit and replace it with an ATtiny45 and a button cell battery.

Get the demo sketch here:

<http://bit.ly/ArduinoISP-material>

Demo circuit (no ISP)



fritzing

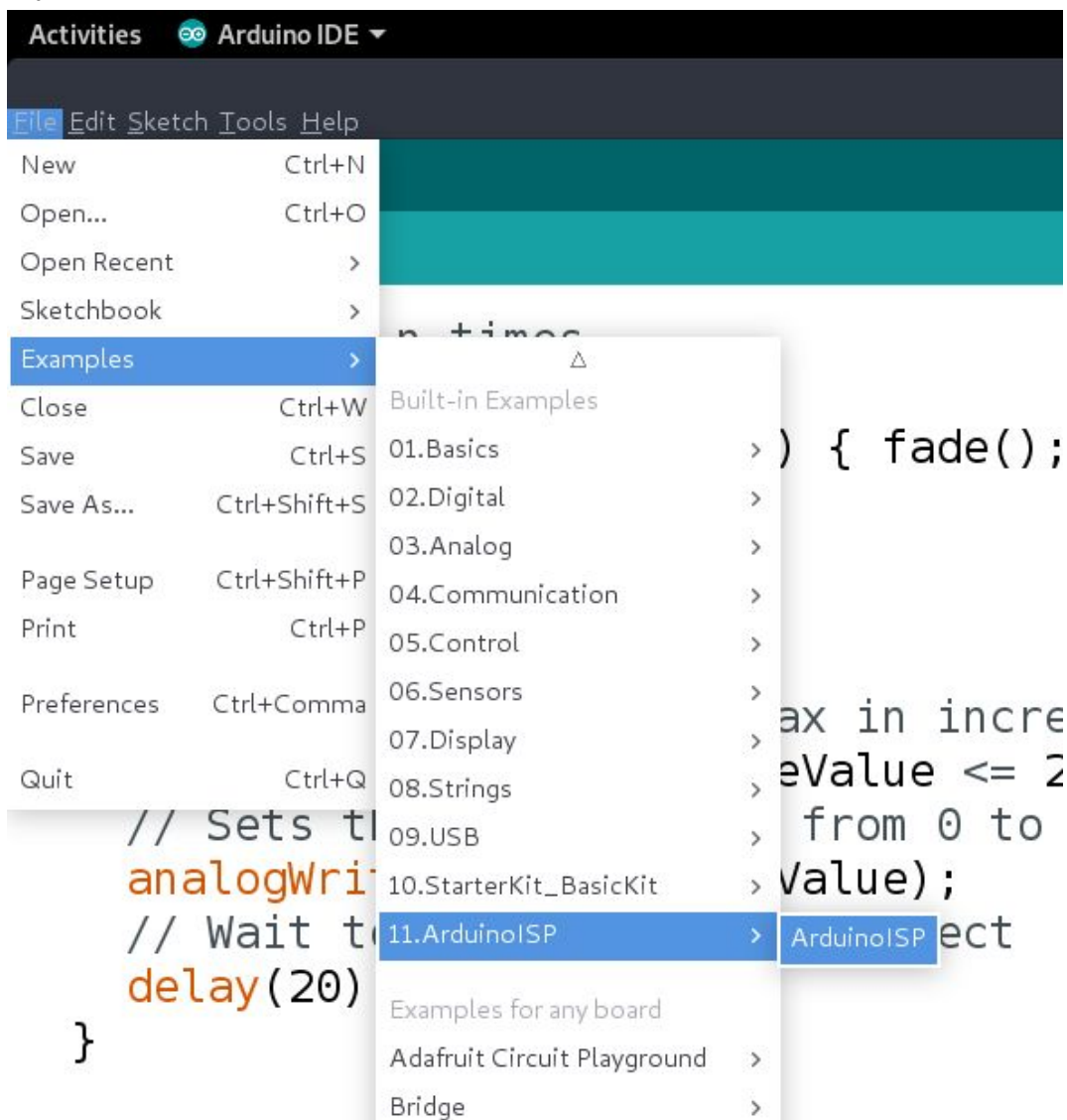
Note: Both resistors are 220 ohm.

Once we are happy with our circuit and sketch prototype, we may proceed to step 2.

Step 2: Uploading the ISP software to the Arduino UNO

This step enables the Arduino UNO to operate as a programmer for other AVR microcontrollers.

Arduino IDE -> File -> Examples -> 11.ArduinoISP -> ArduinoISP -> Upload the ArduinoISP sketch to your Arduino UNO



Upload the ArduinoISP sketch to the Arduino UNO before proceeding to step 2.

From this point we will upload sketches and modifications to the ATtiny45, using the Arduino UNO as a programmer. The Arduino is the interface between our computer and the ATtiny45.

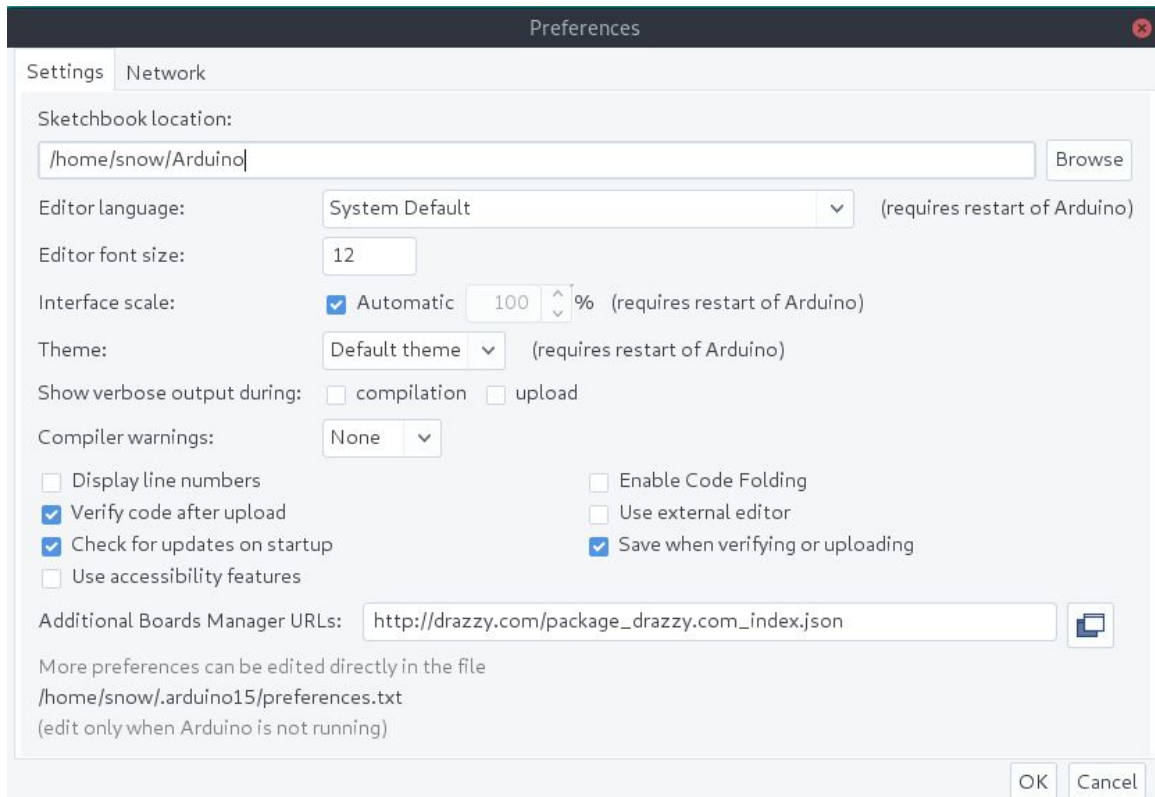
Step 3: Installing ATTinyCore

This step enables IDE support for the ATTiny45 and other AVR microcontrollers.

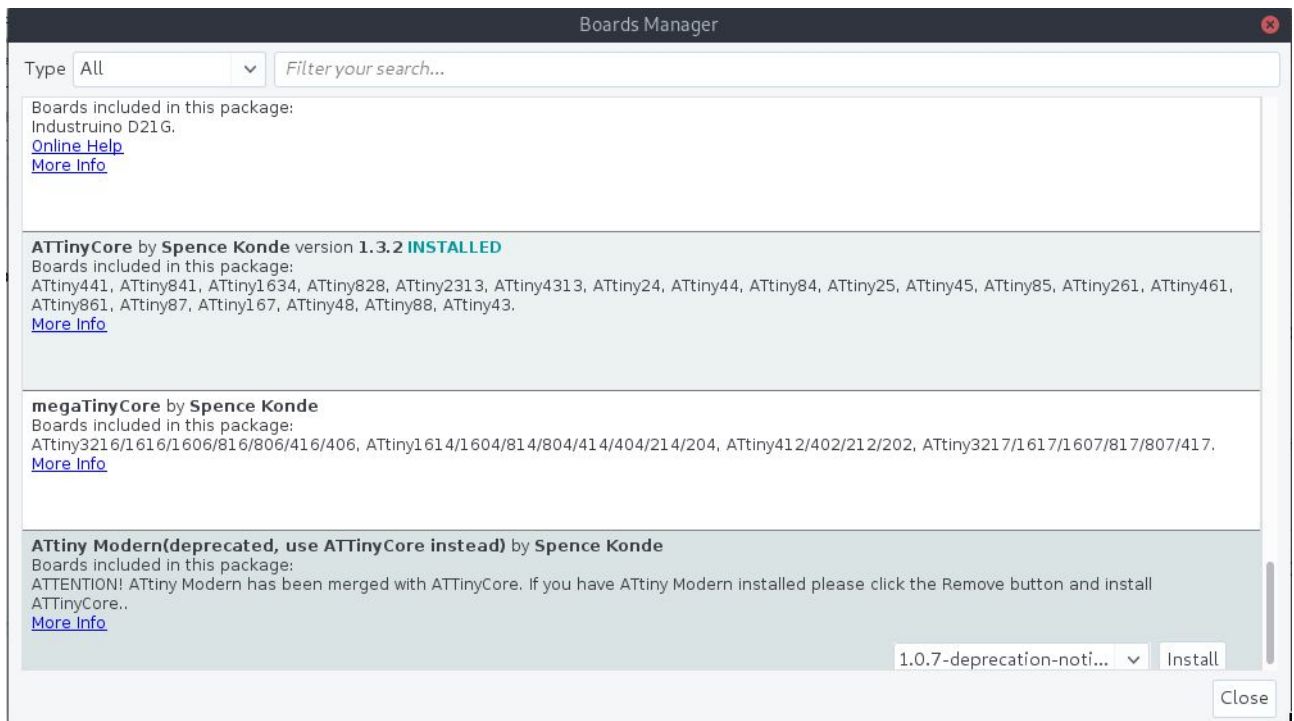
3.1 Add additional boards: Arduino IDE -> File -> Preferences ->

Paste in URL for the additional boards manager:

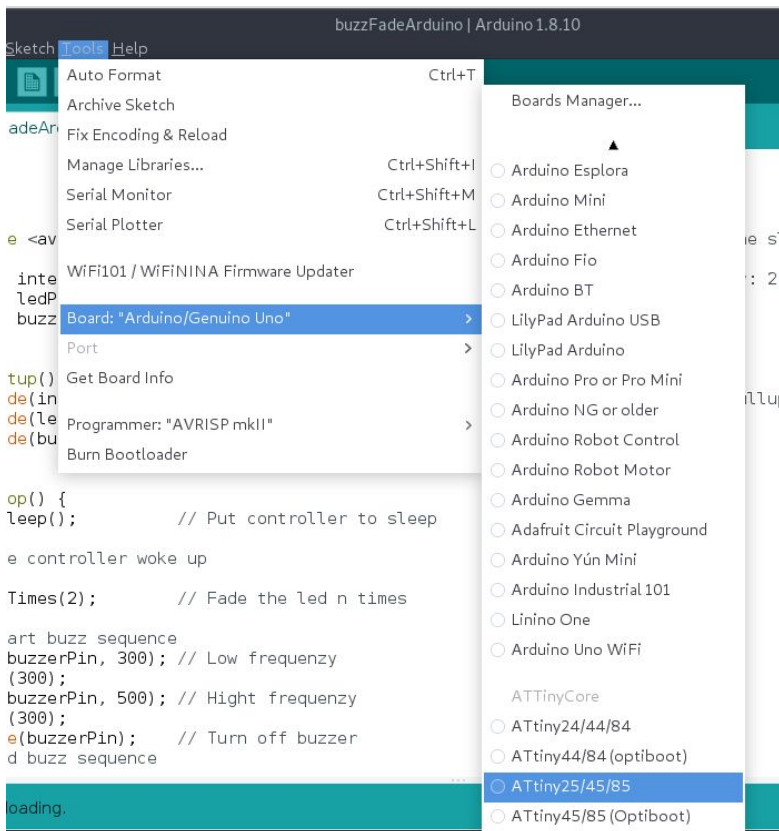
http://drazzy.com/package_drazzy.com_index.json -> OK



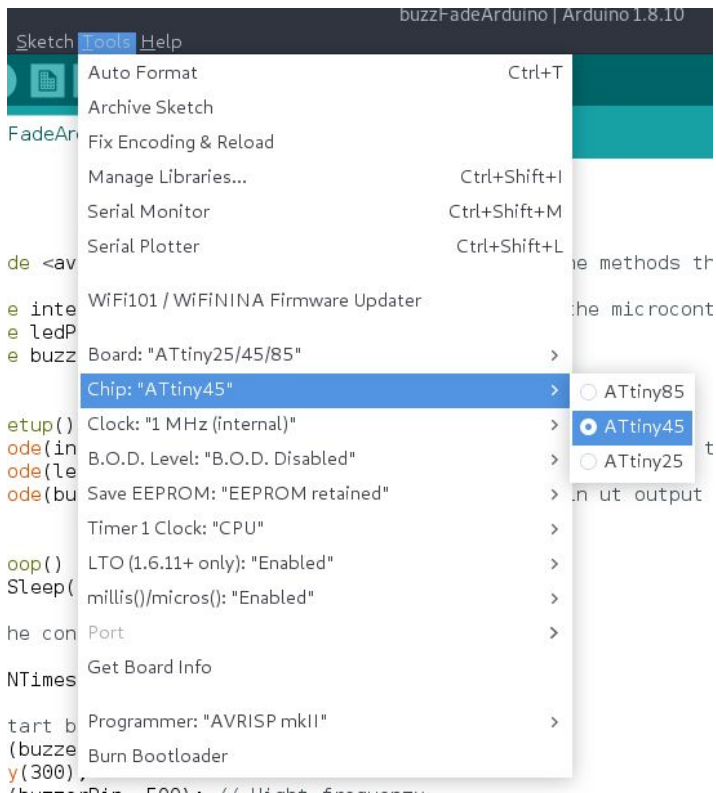
3.2 Install ATTiny Core: Arduino IDE -> Tools -> Board -> Boards Manager -> ATTinyCore -> Install



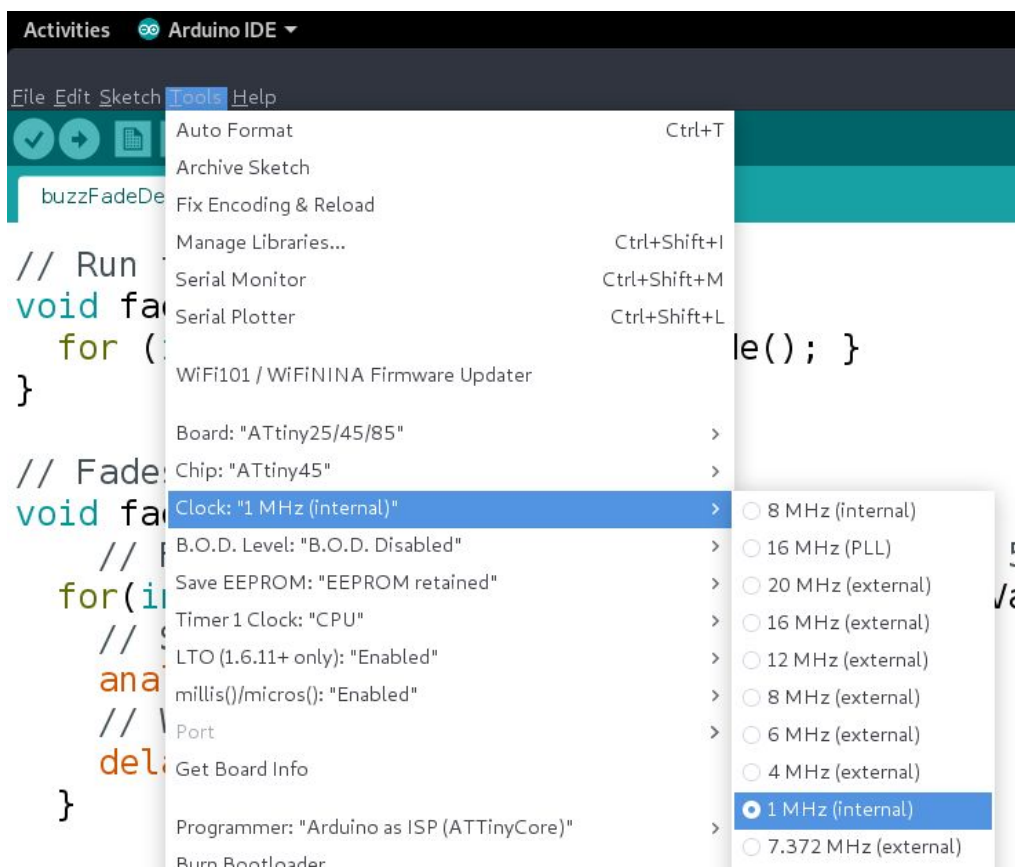
3.3 Select the microcontroller: Arduino IDE -> Tools -> Board -> ATtiny25/45/85



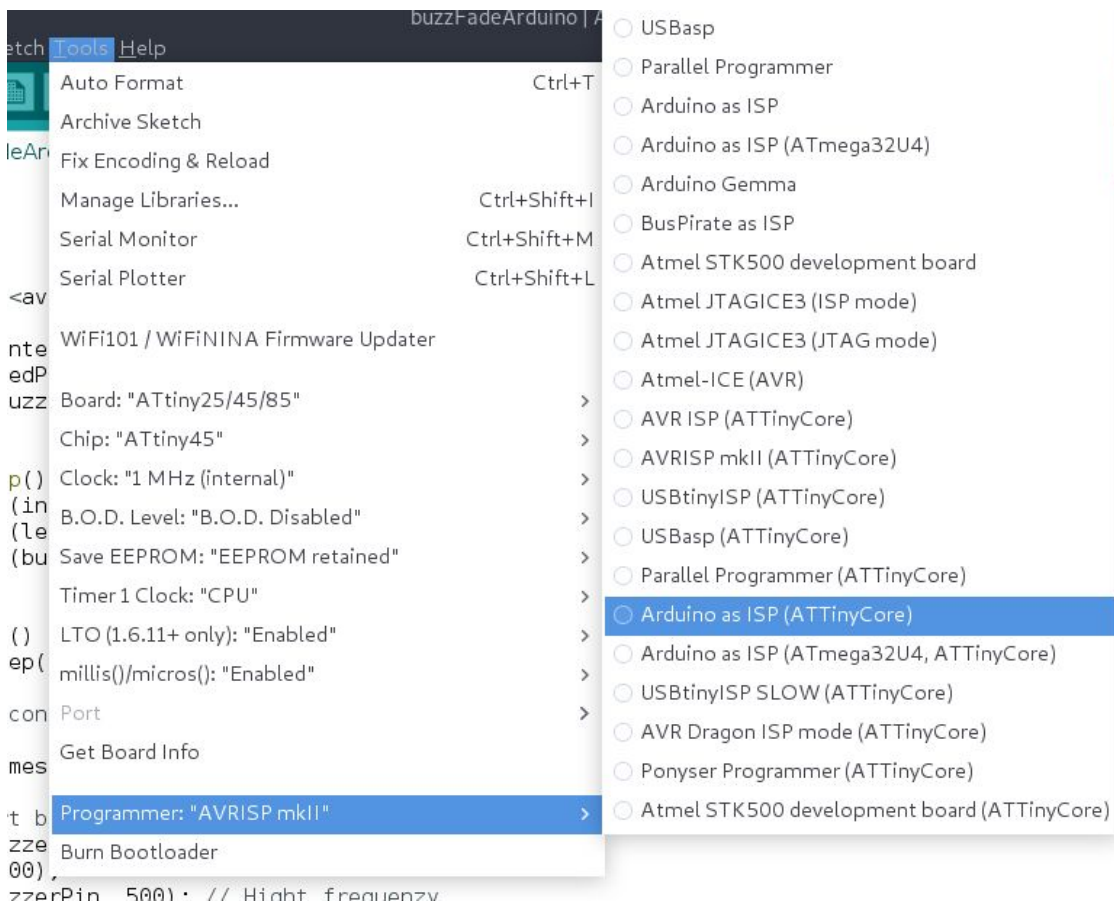
3.4 Select the chip model: Arduino IDE -> Tools -> Chip -> ATtiny45



3.5 Set controller clock frequency: Arduino IDE -> Tools -> Clock -> 1 MHz (internal)



3.6 Select programmer: Arduino IDE -> Tools -> Programmer -> Arduino as ISP (ATTinyCore)

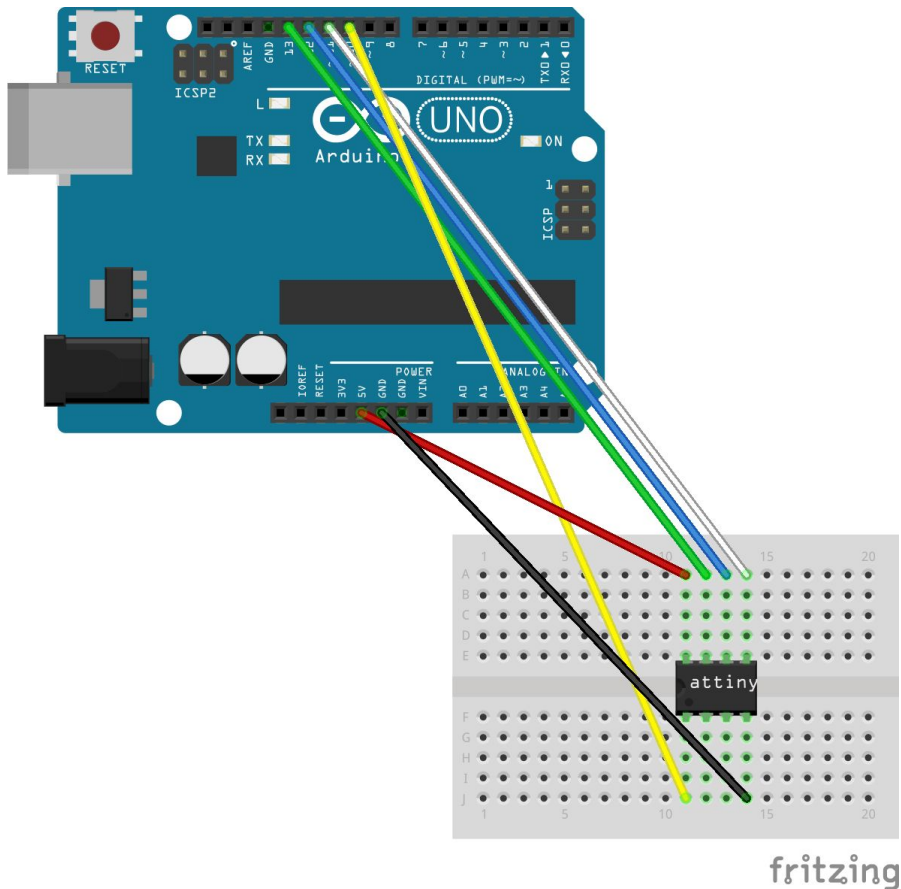


3.7 Set new pin numbers: Arduino IDE -> In demo sketch, change ledPin to 0 and buzzerPin to 1.

Note: Pin numbers for any given sketch and circuit depend on available pins on the microcontroller, available functions for the various pins (specified in the datasheet), and your own preferences.

Step 4: Wiring Arduino UNO as ISP for ATtiny45

This configuration is required whenever you want to program an ATtiny45 using the Arduino UNO as ISP. The microcontroller may be programmed while part of another circuit (in-system programming).



ATtiny45 pinout



Pin connections for ISP

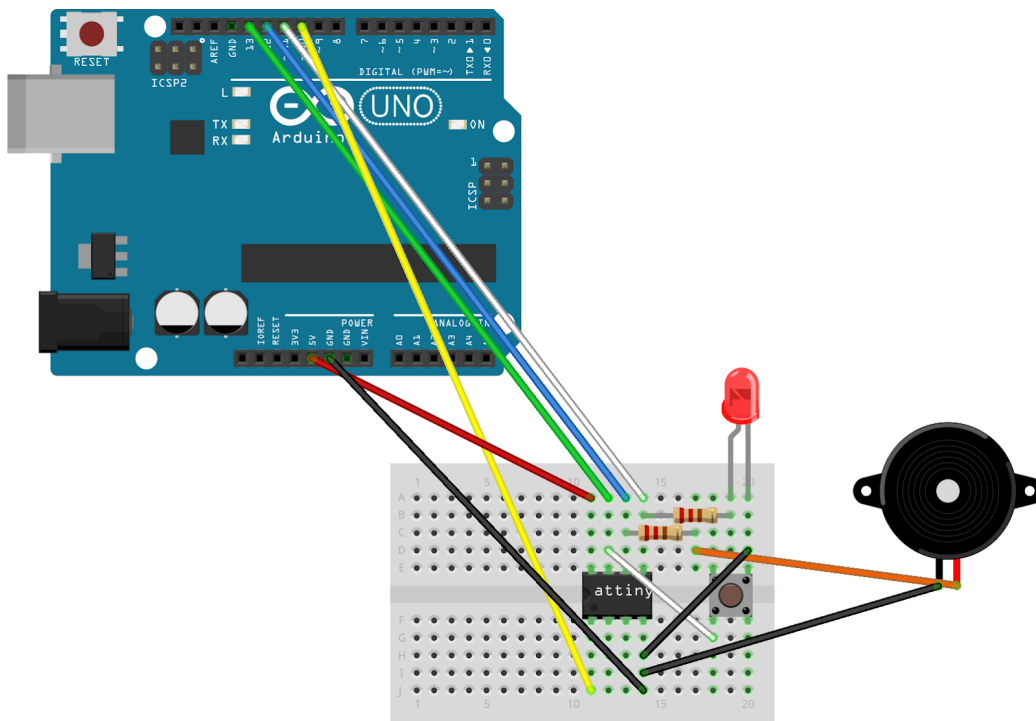
- ATtiny Pin 2 to Arduino Pin 13 (green wire)
- ATtiny Pin 1 to Arduino Pin 12 (blue wire)
- ATtiny Pin 0 to Arduino Pin 11 (white wire)
- ATtiny Reset Pin to Arduino Pin 10 (yellow wire)

Arduino as ISP with demo circuit

4.1 Build circuit with Arduino as ISP

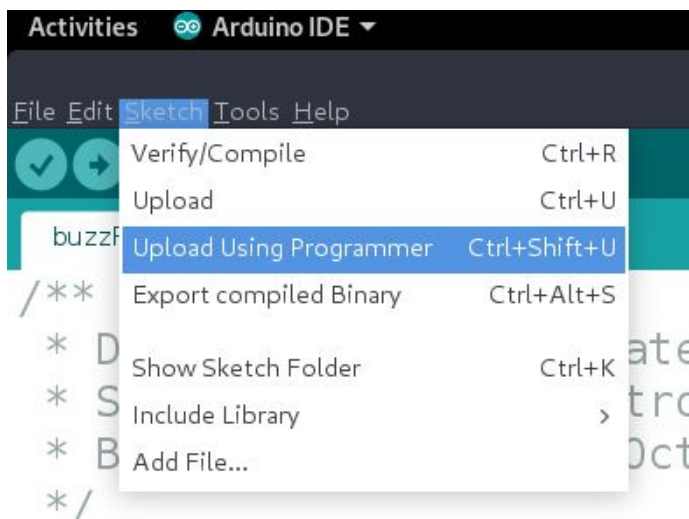
Note: This circuit incorporates the circuit from page 7 (required for programming the ATtiny45) and the demo circuit we built in step 1.

You may modify the sketch and upload while the ATtiny45 is still in the circuit.



fritzing

4.2 Upload the demo sketch to the ATtiny45: Arduino IDE -> Sketch -> Upload Using Programmer



Note: In order to revert to programming the Arduino UNO (no ISP), change board back to Arduino UNO and programmer back to AVRISP mkII via the Tools menu.

Step 5: Final stand-alone demo circuit

Once we are done programming the ATtiny45, we may remove the Arduino UNO entirely. Here the power source is a 3V button cell battery.

