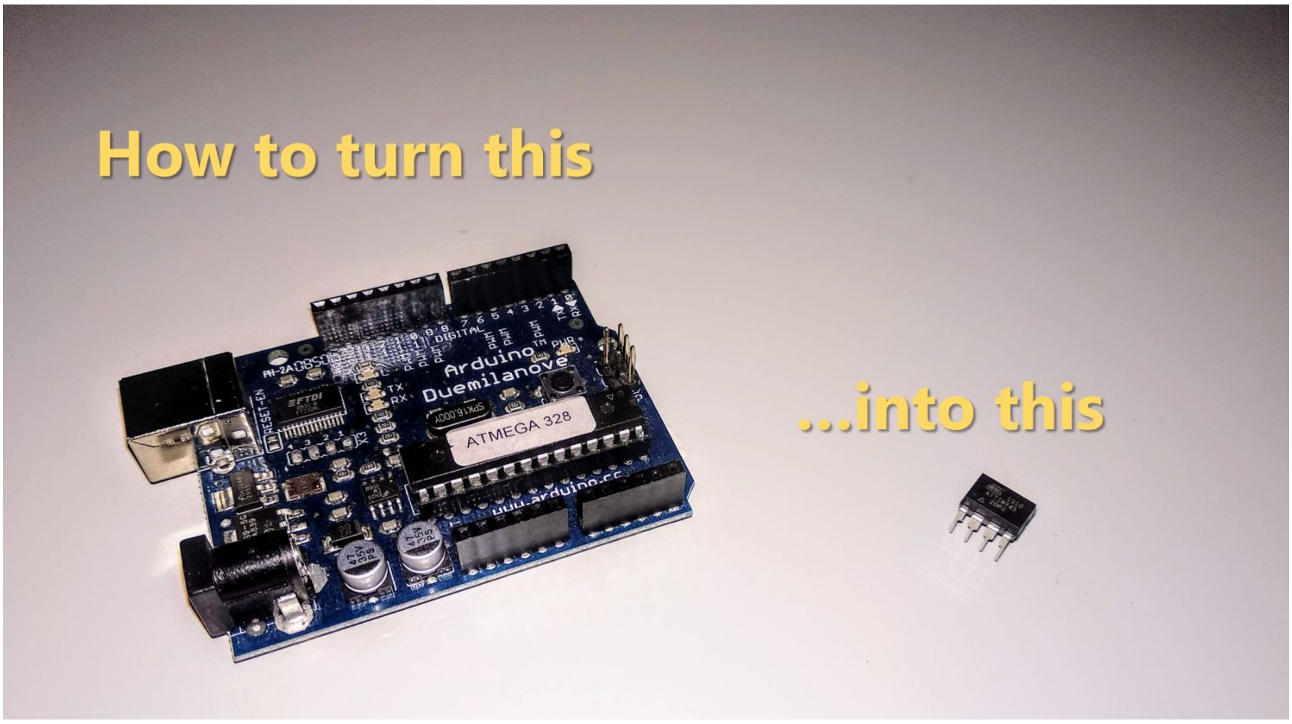
Shrink your microcontroller project using Arduino ISP



Course Abstract

The Arduino has given many makers a low barrier to entry for integrating logic into our projects. Even though the Arduino is remarkably versatile, we sometimes run into limitations. For compact, power efficient and inexpensive projects, or commercial products, using an entire Arduino or single-board computer may not always be practical or economically feasible.

However, for simple logic, such as controlling LEDs, sensors or motors, a smaller microcontroller may be more than powerful enough for the job. A simple microcontroller is typically only a few square millimetres, consumes very little power (from a few milliamps when operating to a few microamps when sleeping), and cost only a few øre to a few kroner.

Luckily, taking the first step into AVR microcontroller programming is relatively easy if you already have experience with Arduino.

The heart of the Arduino is itself an AVR microcontroller (the ATmega328) which shares its basic architecture with other AVR microcontrollers. This enables sketches written for the Arduino to run on much less expensive, smaller, and more power efficient AVR microcontrollers with relatively few modifications.

Additionally, free software enables you to turn your existing Arduino into an in-system programmer (ISP) for other AVR microcontrollers, so you won't have to invest in expensive gear or learn new software in order to get started.

Arduino ISP in a nutshell

Essentially, you write your sketch for the Arduino and begin your basic testing and prototyping. You then do the necessary rewriting of the sketch for the AVR microcontroller and load the ISP programme onto your Arduino. After wiring the microcontroller to the Arduino, you transfer your sketch from the IDE on your computer to the microcontroller. The simpler microcontroller can now replace the Arduino in your original circuit.

What we will cover:

- Basic understanding of microcontrollers.
- Using the Arduino as an in-system programmer (ISP).
- Transfer simple Arduino programs from the IDE to an AVR microcontroller (we'll use the ATtiny45).
- Running our circuit and microcontroller on button cell batteries.

Prerequisites:

- You have written or modified simple Arduino programs before and have a basic understanding of programming concepts such as variables, loops, and functions.*
- You bring your own computer with Arduino IDE installed (version 1.8 or later)

*The organiser will provide all the code necessary to complete this course.

Tools and components:

Participants may bring their own components or borrow from the organiser.

- Computer with Arduino IDE installed (bring your own)
- Arduino UNO (most clones and other Arduinos with ATmega328 work as well)
- USB cable to connect the Arduino UNO to your computer
- ATtiny45 microcontroller
- 3V CR2025 button cell battery
- Battery holder or heat shrink tubing (wide enough for the button cell)
- Breadboard
- Jump wires
- Two 220 Ω resistors (close values work as well)
- Two-pin LED
- Four-legged push-button
- Piezo buzzer

Agenda:

1. Welcome and introduction to the course.
2. Basic demonstration of ISP programming (what we'll be able to do by the end of the evening).
3. What is a microcontroller?
4. What is in-system programming?
5. Let's wire our Arduino prototype circuit and load the example sketch (push the button and enjoy the fading LED).
6. Explaining the ATtiny45 pinout.
7. Installing ATtiny support in the Arduino IDE.
8. Modifying our sketch to work with the ATtiny45.
9. Loading the ISP programme onto our Arduino UNO.
10. Wiring the Arduino UNO as an ISP for the ATtiny45.
11. Transferring our modified sketch to the ATtiny45.
12. Inserting the ATtiny45 into our prototype circuit, remove the Arduino, and attach the battery.
13. Marvel as our circuit continues to do its magic without the Arduino.
14. If time: Reading the microcontroller datasheet and setting registers to access controller-specific features, such as sleep functions.
15. What is the next step if you which to do more advanced stuff with microcontrollers?