
Embedded Programming with Arduino

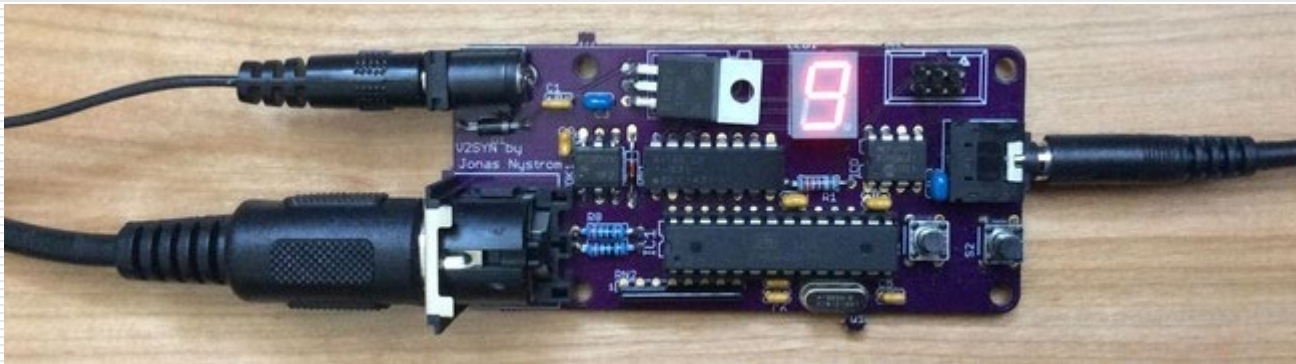
Rodney Dorville

Summary

- What is embedded programming?
 - Arduino Systems
 - Arduino Boards and Atmel processors
 - The Arduino IDE System
 - Writing your first sketch
 - Blink
 - Digital output & PWM
 - Digital input
-

Embedded Programming

- A dedicated computer system with a dedicated function within a larger system to perform a specific task. E.g. smart TVs, ovens
- After development with the Uno, the circuit could be reduced to a single IC with support devices.



Ref: [MatrixSynth – ATmega328 synthesizer](#)

What is an Arduino?

- An Arduino System comprises of
 - Software & Software Tools
 - ❖ Integrated Development System (IDE)
 - ❖ Arduino programming language (C++ like) based on Wiring
 - ❖ Development & Debugging tool based on Processing
 - ❖ Software libraries (open source contributions)
 - Hardware
 - ❖ ATMel processor board
 - ❖ Shields (Add-on modules)
 - ❖ Sensors, actuators, peripherals
- Open Source Platform

Ref: [Open Source Software](#)

Arduino (common misconceptions)

- Made up of a microprocessor board (Arduino Uno)
 - Can be used to build projects (with the board)
 - Programmed in C++
 - Has lots of clones, which may be incompatible

 - Arduino is **more** than that (see later)
-

Arduino Systems

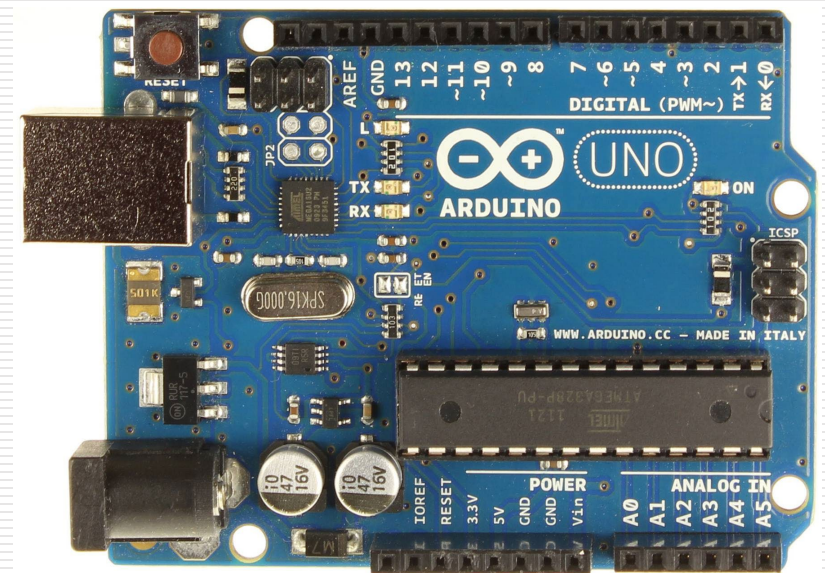
- A hardware and software company based in Italy
 - Initial development – software (based on Wiring) and later hardware boards (Atmel based)
 - Produces and markets “official” boards: Uno, Due, Leonardo, Diecimila, Mega, Nano
 - Software and hardware is open source.
-

Why use Arduino?

- Inexpensive (cost < \$5)
- Cross-platform (Mac, Linux, Windows)
- Simple, clear programming environments using a GUI
- Open source and extensible software
- Open source and extensible hardware

Arduino Uno

- Most common microcontroller board to begin Arduino projects.
- Uses a Atmel Atmega328P processor with a separate programmable interface using another Atmel processor and USB.
- Has sockets for interfacing and power.



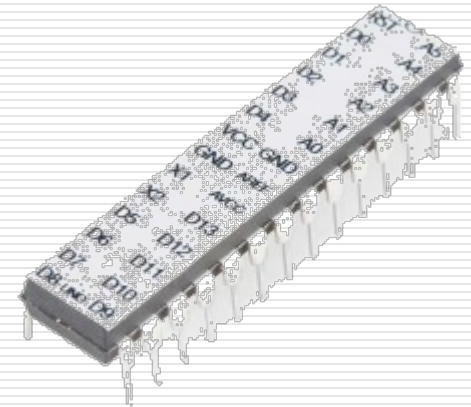
Alternative?

- Learn C++ with gcc compilation
 - Setup an Atmel toolchain
 - C++ compiler (to write your project)
 - AVR compiler and toolchain (to convert C++ to asm)
 - Calculate the Atmel AVR fuses, and set them for your system
 - Use AVRDude to program the Atmel processor
 - Test using electronic test tools

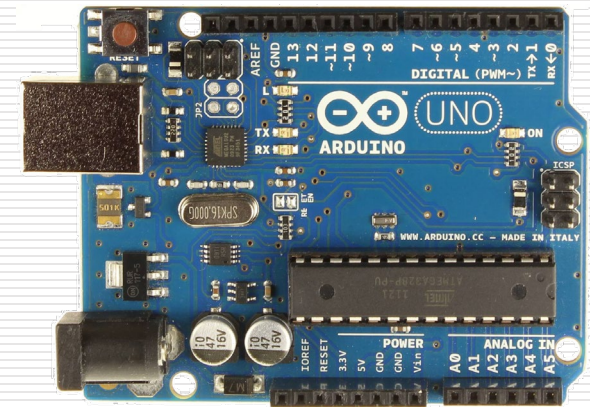
 - Difficult, time consuming, requires expert knowledge and experience.
 - Not for the beginner, but for more precise control.
-

Arduino ATmega328P processor

- Microcontroller ATmega328P
 - ❑ 8-bit AVR processor running at 16MHz
 - ❑ 32K Flash RAM, 1KB EEPROM, 2K SRAM
 - ❑ 14 digital input/output pins (6 can be used as PWM)
 - ❑ 6 analog inputs (10-bit ADC)
 - ❑ Serial USART
 - ❑ SPI, I2C, Timer, comparator
 - ❑ 2 8-bit timers, 1 16-bit timer, RTC
- UNO adds
 - ❑ Second processor for USB interface
 - ❑ Interface sockets for easy use
 - ❑ Separate power and reset



Labelled ATmega328P processor



UNO board

Ref: [Atmel ATmega328 datasheet](#)

Arduino System & Boards

- [Youtube: MakeUseOf – Thinking about Getting an Arduino](#)



Embedded Programming with Arduino

Rodney Dorville
