EXPERIENCES FROM AN ARDUINO PROGRAMMING COURSE FOR TEACHERS

Đorđe Herceg°, Dejana Herceg *

° Faculty of Science, University of Novi Sad

* Faculty of Technical Sciences, University of Novi Sad

The authors

- Djordje Herceg, professor at the University of Novi Sad Faculty of Science, Mathematics and Infomatics
- Dejana Herceg, professor at the University of Novi Sad Faculty of Technical Sciences (Engineering)

- 20+ years of teaching experience at the University
- 15 years at the secondary school
- Courses for teachers and industry
- Software projects for the University and other customers

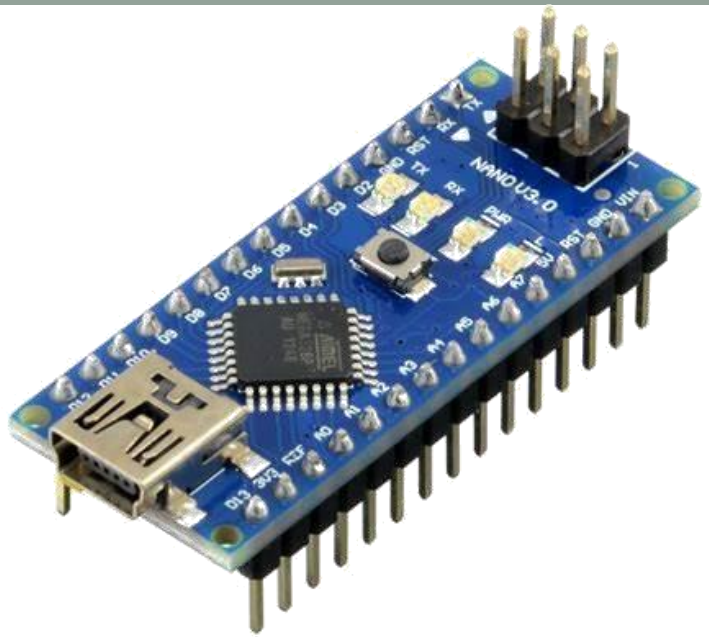
Motivation

- Introduced an undergraduate course on Arduino and IoT programming at the Faculty of Science, Department of Mathematics and Informatics
- Students have little to no experience with electronics
- In Computer Organization and Architecture they learn about the operation of digital computers and assembly
- However the IT industry in Novi Sad offers jobs in IoT
- Our course connects CO&A with high level programming and also offers introduction to solving real life problems
- The main challenge is to deliver a completed product

What is Arduino?

- Open-source electronics platform based on affordable and simple hardware
- Microcontroller CPU with built-in memory and I/O ports
- Programmed in C++ or other languages
- Arduino IDE
- Arduino framework
- Software libraries
- Hardware devices and add-on boards

Hardware specs



- Arduino Uno
- 8-bit Atmel AT328 @ 16MHz
- 32KB program Flash
- 2KB RAM
- 1KB EEPROM
- 23 general purpose I/O pins, some with PWM
- 6 channel A/D converter
- UART, SPI and I2C interfaces

Arduino-based learning kits

- Affordable and available in many configurations
- Individual components can be obtained directly from the source at lower prices

Target audience

- We developed a curriculum on C++ programming for Arduino with introduction to electronics
- Target audience: elementary and middle school teachers
- Three 16-hour courses held in 2018
- 47 participants, 23 male and 24 female
- 8 elementary school teachers
- 39 middle school teachers
- 16 participants with 8-15 years of experience
- 19 participants with 16-25 years of experience

Course outline

- Analog and digital electronics
- Arduino hardware and related devices
- Arduino software
- C++ programming
- Specific C++ topics: pointers and memory access
- Advanced topics: registers, interrupts, multitasking, interprocess communication, serial interfaces
- Encapsulation, abstraction and sepraration of concerns: enums, structs, clases, drivers, libraries

Aims and goals

- **Demistify working principles of digital technology**
  (finally explain what those ones and zeros are all about)
- **Teach basics of machine-level programming**
  (as opposed to high-level JavaScript frameworks)
- **Emphasize the importance of algorithmic thinking**
  (one thing at a time)
- **Teach patience and planning**
  (instead of immediately typing in code)
- **Explain the evolution of computers**
  (did you know that keyboard is considered a ’luxury’?)

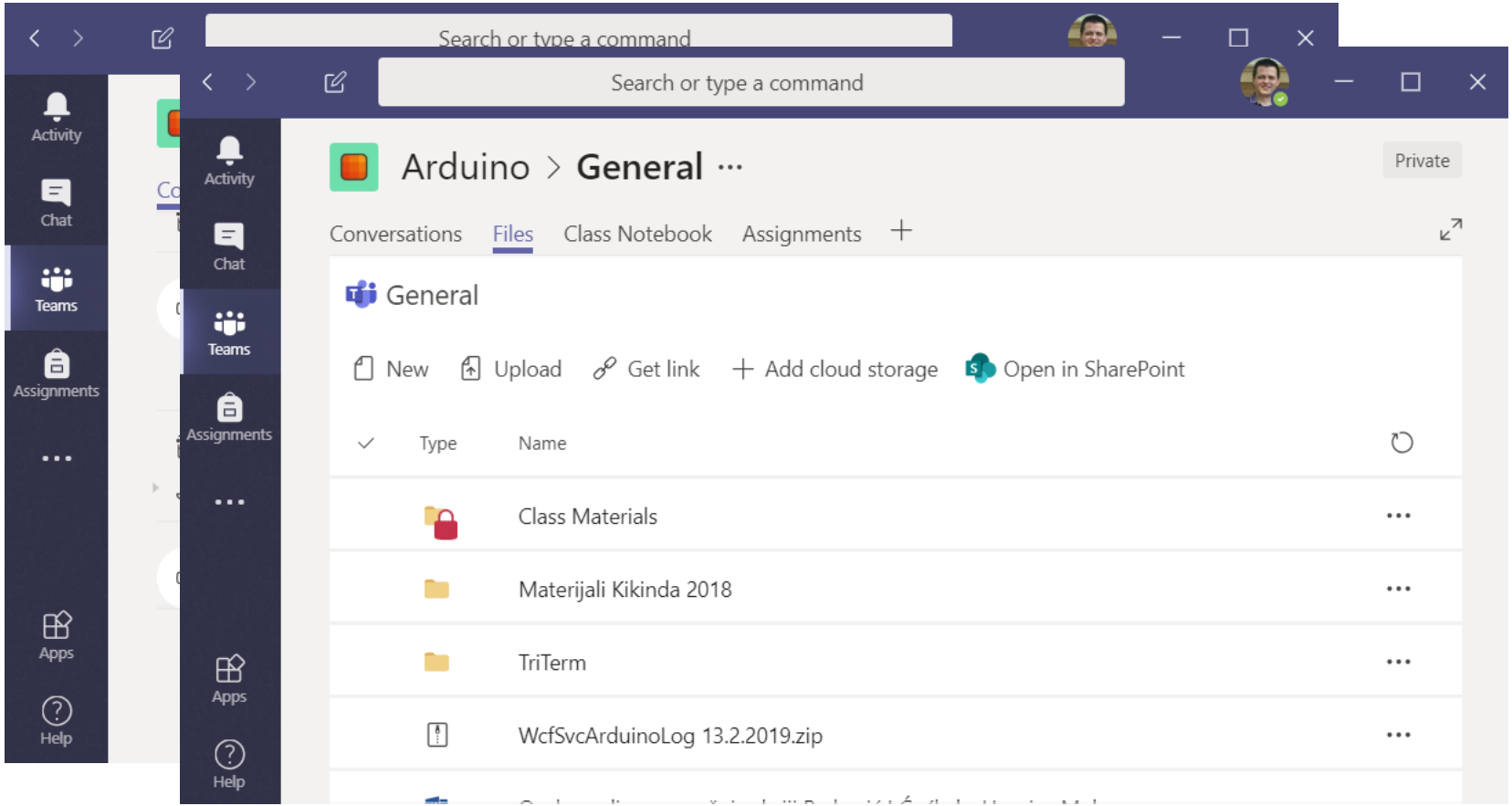
Prerequisites

- Some programming experience is needed
- Basic knowledge of electronics -and mathematics
- A complete shift in programming paradigm is necessary as soon as non-trivial examples are introduced
- Smart houses? Intelligent robots?
  Yes, but that will have to wait...
  ... at least until you learn how to read a keyboard!
- A modest computer will do for the Arduino IDE (Windows, Linux, MacOS)

Office 365 for content delivery

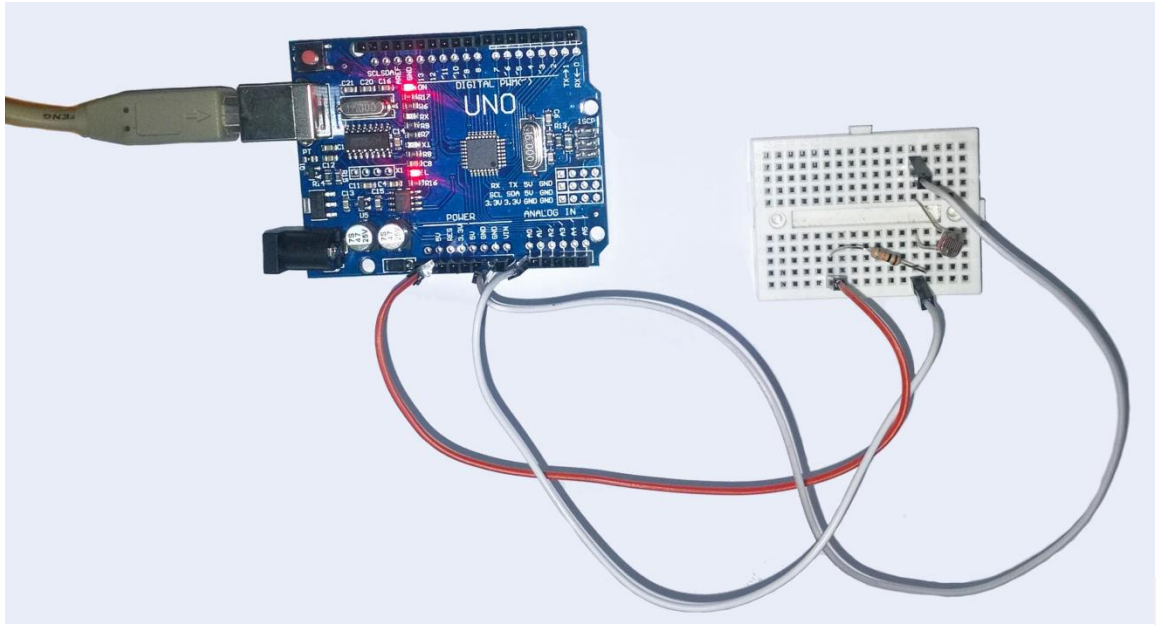
- Teaching materials prepared beforehand (PDF)
- Circuit diagrams
- Example programs
- References to relevant web sites

- Office 365 logins for all participants
- Microsoft Teams for screen sharing and communication
- SharePoint site for assignment collection


Search or type a command
Activity
Chat
Teams
Assignments
Apps
Help
Search or type a command
Activity
Chat
Teams
Assignments
Apps
Help
Arduino > General ···
Private
Conversations
Files
Class Notebook
Assignments
General
New
Upload
Get link
Add cloud storage
Open in SharePoint
Type
Name
Class Materials
Materijali Kikinda 2018
TriTerm
WcfSvcArduinoLog 13.2.2019.zip

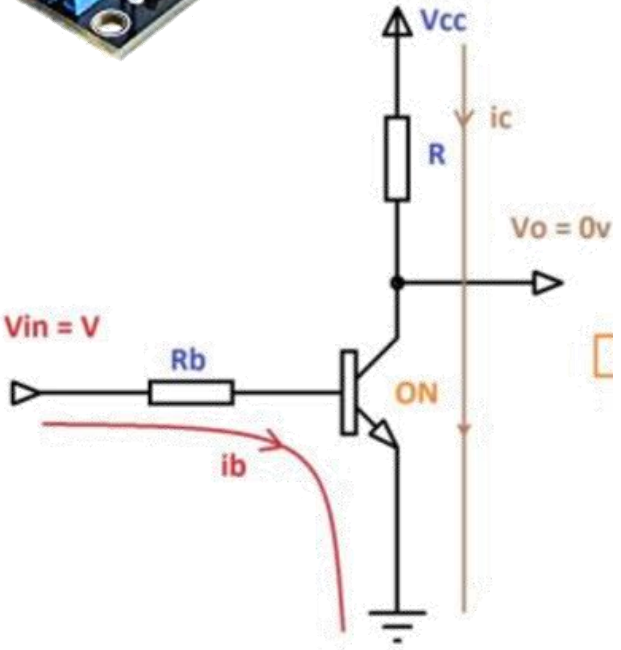
Basic examples

- Blinking LED
- Light sensor
- Buzzer
- Digital input
- Traffic light
- Potentiometer
- Serial data



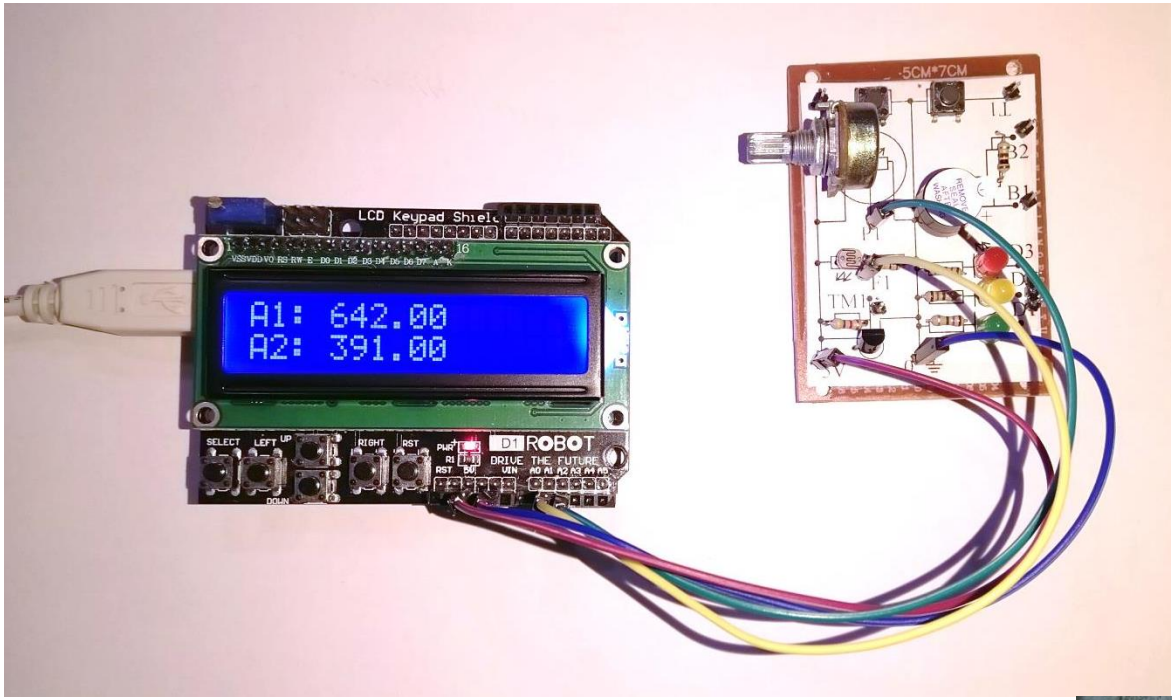
Example – light sensor

- setup()
  - Initialize the light sensor
  - Initalize the relay
- loop()
  - Read the light sensor value
  - Check if the value is over a threshold
  - If yes, turn the relay on
  - If no, turn the relay off
  - Optionally, display status on a LED or LCD
  - Wait a short time
  - Repeat



Complex examples

- Libraries
- LCD screens
- Keypad reading
- Countdown timers
- State machines
- User interface considerations
- RTOS



Gradual release of responsibility

- From „I do“ to „We do“ and finally „You do“
- First the teacher is demonstrating the desired outcome
- Then the teacher and the pupils collaborate
- Finally the pupils work independently
- Individual work in pairs and small groups

- Requires independence, maturity and self-confidence
- Knowledge of the English language is a must
- Browsing the Web and filtering the results

Experiences from the course

- The teachers were well motivated
- The course alternates between lectures and exercises
- We prepared Arduino kits for hands-on exercises
- Solutions to problems were prepared beforehand or demonstrated after the exercises and handed out later
- Depth-first approach: focusing on important parts
- We assisted during the exercises

Observations

- Most teachers were satisfied with the topics and scope of the course
- Some found C++ difficult
- For some, the programming model was counter-intuitive because it doesn't follow the linear program structure
- Abstractions are not visible from the code
- Problems with wires falling out, reversed polarities and libraries for hardware components
- No debugger!

Observations (cont'd)

- Some struggling with elementary calculations (current, resistance and similar)
- Difficulties compiling the known parts into a working whole
- Debugging the software is difficult, but not impossible, thanks to Serial.print
- Debugging the hardware is more difficult

- Real life problems do not conform to carefully prepared classroom exercises! („the numbers aren't nice")

Conclusions

- Most of the teachers are competent and curious enough about Arduino and the possibilities it brings
- However some had difficulties adapting to a different programming model and abstractions which are not explicitly formulated in hardware or software
- Fear and lack of confidence hinder their performance

- Arduino connects real life to the classroom, but doesn’t shield the user from the complexities of computers

- Plan for failure and learn from it

Conclusions (cont'd)

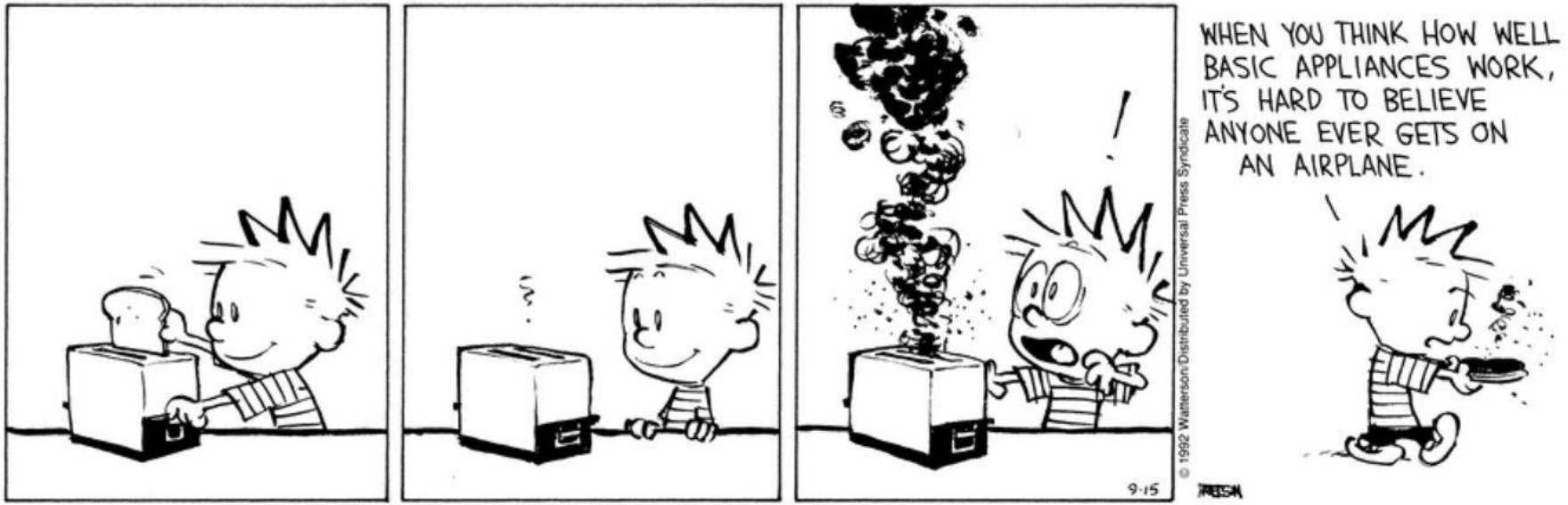
- Attention should be devoted to confidence building exercises for teachers
- Lifelong learning is not just a catchy phrase, it is a necessity
- Failed exercises should not be feared, they should be planned for and learned from
- Curiosity is a positive trait
- Strong emphasis on responsibility
- Double the hours or halve the curriculums!

Plans for the future

- Develop more learning materials
  - a book
  - Videos
  - Office 365 resources

- Teach software project management in practice
- Connections to IT industry, joint projects for students
- Academic work that stems from professional work
- Networking

Calvin and Hobbes by Bill Watterson



Thank you!