ARDUINO
family of boards
for Internet of Things - IoT

Vladimir Cvjetkovic
Faculty of Science
University of Kragujevac
What is IOT?

• IoT consists of many small computing devices capable of interacting with environment and which are at the same time connected to network

• Interaction with environment means that devices can measure some physical characteristics that are important, and that can also act upon the environment in order to change some environmental physical characteristics.

• Network connection enable both transfer of data and commands for initiation of various actions

• IoT is very convenient to be implemented with various Arduino boards that can be programmed and interconnected in various ways
ARDUINO members

- General purpose ARDUINO boards
  - MCU only boards
  - Combined MCU / MPU boards
- Special purpose ARDUINO boards
  - ARDUINO Esplora
  - ARDUINO Robot
- ARDUINO compatible boards
  - Intel Galileo, Gen 2
  - Intel Edison
- ARDUINO shields
  - Ethernet, WiFi, GSM, Motor, Relay and others
MCU versus MPU

- MCU (Micro Controller Unit)
  - Real time predictable
  - Self contained
  - Limited memory
  - Used for embedded tasks
  - Price: CHEAP

- MPU (Micro Processor Unit)
  - No real time
  - Not self contained
  - Limitations less strict
  - Used for general purpose
  - Price: EXPENSIVE

- MCU & MPU are complementary!
General purpose MCU only boards

- ARDUINO UNO - Atmel AVR ATmega 328
- ARDUINO Ethernet - Atmel AVR ATmega 328
- ARDUINO Leonardo – Atmel AVR ATmega 32u4
- ARDUINO Leonardo ETH – Atmel AVR ATmega 32u4
- ARDUINO Mega – Atmel AVR ATmega 2560
- ARDUINO M0 – Atmel ATSAMD21G18 based on ARM Cortex M0 processor
- ARDUINO Due - Atmel SAM3X8E based on ARM Cortex-M3 CPU
General purpose MCU & MPU combined boards

- ARDUINO Yun – ATmega 32u4 & Atheros AR9331 processor – Linux, WiFi, Ethernet, microSD card, USB host

- ARDUINO Tian - Atmel SAMD21 MCU, based on 32-bit ARM Cortex® M0 core & Qualcomm Atheros AR9342, which is a highly integrated MIPS processor - Linux, WiFi, Ethernet, Bluetooth, USB host
Arduino YUN Bridge
What a general purpose board can do?

- Elementary tasks
  - Measurement and generation of analog voltages
  - I/O operations on digital pins
- Number of analog & digital pins may vary
- More complex tasks can be broken down to elementary tasks
- Communication, measurements with various sensors, management of Ethernet, WiFi, LCD and other complex tasks are supported by a number of easy to use libraries
- Arduino is open source both hardware & software
<table>
<thead>
<tr>
<th>Board / features</th>
<th>Uno</th>
<th>Due</th>
<th>Galileo</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>6, 10 bits</td>
<td>12, 12 bits</td>
<td>6, 12 bits</td>
</tr>
<tr>
<td>AO</td>
<td>6, PWM, 8 bits</td>
<td>2, DAC, 12 bits</td>
<td>6, PWM 8 bits</td>
</tr>
<tr>
<td>DIO</td>
<td>14, 6 PWM, 8 bits</td>
<td>54, 12 PWM 8 bits</td>
<td>14, 6 PWM 8 bits</td>
</tr>
<tr>
<td>Processor</td>
<td>ATMega 328</td>
<td>AT91SAM3X8E</td>
<td>Intel Quark SoC X1000</td>
</tr>
<tr>
<td>Clock</td>
<td>16 MHz</td>
<td>84 MHz</td>
<td>400 MHz</td>
</tr>
<tr>
<td>Flash</td>
<td>32 KB</td>
<td>512 KB</td>
<td>8 MB / 512KB</td>
</tr>
<tr>
<td>SRAM</td>
<td>2 KB</td>
<td>96 KB</td>
<td>512 KB</td>
</tr>
<tr>
<td>DRAM</td>
<td>-</td>
<td>-</td>
<td>256 MB</td>
</tr>
<tr>
<td>EEPROM</td>
<td>1 KB</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Micro SD</td>
<td>- (on shield only)</td>
<td>- (on shield only)</td>
<td>Up to 32 MB</td>
</tr>
<tr>
<td>Ethernet</td>
<td>- (on shield only)</td>
<td>- (on shield only)</td>
<td>10/100 Mb/s</td>
</tr>
</tbody>
</table>
Serial communication among boards

- **USB** using **UART** (Universal Asynchronous Receive Transmit) with PC for development purposes
- **UART** Hardware / software among two boards using digital I/O pins
- **TWI** (Two Wire Interface) **SDA** / **SCL** pins (Serial Data / Serial Clock)
- **SPI** (Serial Peripheral Interface) using 4 pins **MISO** (Master Input Slave Output), **MOSI** (Master Output Slave Input), **SCLK** (Serial Clock), **SS** (Slave Select)
- **Network** connection with **Ethernet** / **WiFi** / **GSM** built in or shields – extensions
- **microSD** cards – copy on one device, read on other
Why to connect boards?

- Arduino boards are similar to Lego building blocks.
- Not very much can be done with only one block, but with many blocks one has almost unlimited possibilities.
- Processing power or some other requirements may exceed capacities and resources of a single board.
- In that case, joined resources of two or more connected boards may solve the problem.
- Single board or system of boards can be connected to other larger computer systems.
Various ways of board programming

- Programming from Arduino IDE in Arduino Language (AL) similar to C++
- Programming in Java Script
  - NodeJS with JohnnyFive library on PC using Firmata protocol on board
  - NodeJS with JohnnyFive library using Galileo-IO on board from Linux
  - Libmraa library on board from Linux (Galileo board)
  - NodeJS on server with web interface to ARDUINO board with network adapter
  - Arduino Yun has Bridge library that connects MCU with MPU
- Programming in other languages - Python from Linux which communicates with a board through some serial wire interface or network
Arduino board connected with PC

- Arduino IDE
- Arduino sketch
- USB
- Arduino board
- Compiled code
Arduino programming with JS

NodeJS

Johnny-Five

USB

Arduino board

Arduino Firmata
Program development with Sysfs

SSH client
Program files
Internet
Galileo
USB COM
Linux Sysfs
Configuration and reading of A0 with Sysfs

```bash
login as: root
root@galileo:~# echo -n "37" > /sys/class/gpio/export
root@galileo:~# echo -n "out" > /sys/class/gpio/gpio37/direction
root@galileo:~# echo -n "0" > /sys/class/gpio/gpio37/value
root@galileo:~# cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
711
root@galileo:~# cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
351
root@galileo:~# cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
926
root@galileo:~# ```
Program development with JF / Galileo-IO
Program development with Intel XDK IoT IDE and mraa library
Single board with Ethernet adapter

- Arduino IDE
- Arduino sketch
- Arduino board
- Compiled code
- Ethernet shield
- USB
- SPI
- Internet
- CLIENT
Arduino with web client access

- Arduino IDE
- Arduino board
- Ethernet shield
- Arduino sketch
- Compiled code
- USB
- SPI
- Internet
- Node.js
- Client
Arduino board with Firmata and Johnny-Five
Scaling of measurement and control system

- Main web server
- Level 1
  - Web server 1
  - Level 2
    - Web server 1
    - Acquisition server A
    - Acquisition server B
    - User
- Level 3
  - Web server 1
  - Acquisition server A
  - Web server 1
  - Acquisition server A
  - User
- Level 2
  - Web server 1
  - Acquisition server A
  - Web server 1
  - Acquisition server A
  - User
- Level 3
  - Web server 1
  - Acquisition server A
  - Web server 1
  - Acquisition server A
  - User
ARDUINO application for temp. & rel. hum. & RC exp.

RC circuit experimental setup
Лаб за Информатику

Температура: 20.5°C
Релативная влажность: 29.3%
RC circuit experiment

Click anywhere to go to experiment
3 connected ARDUINO cards
Web page controlling Esplora

Arduino Esplora sensors & actuators

**Sensor:** Accelerometer

Temperature C: 23 F: 73

Light: 958

Microphone: 1

Slider: 512

Button Right: 1 Left: 1 Up: 1 Down: 1

Joystick Jx: -4 Jy: 9 Jb: 1

**Accelerometer** Ax: 22 Ay: 41 Az: 172

**RGB LED color**

Red: 67

Green: 36

Blue: 204

**Set LED color**

Signal: 97

Pause: 1000

**Set tone**

Tone: 254

Click anywhere to go to experiment
Esplora sensors and actuators
Acronyms from previous slide

• A – Accelerometer
• B – 4 Buttons (on / off) up, down, left, right
• J – Joystick
• L – Light sensor
• LS – Loudspeaker
• M – Microphone
• MCU – Micro Controller Unit
• RST – ReSeT button
• RGB – Red Green Blue LED diode
• Rx/Tx – Serial connectors (UART) – Receive / Transmit
• SL – SLider – potentiometer
• T – temperature
• USB – USB connector for power supply & programming