

Курс “Практическа роботика
и умни “неща” - 2020

Роботика и Интернет на “нещата” (IoT)

Trayan Iliev

tiliev@iproduct.org
<http://iproduct.org>

About me



Trayan Iliev

- Lecturer at Sofia University
- CEO of IPT – Intellectual Products & Technologies
- Oracle® certified programmer 15+ Y
- end-to-end reactive fullstack apps with Java, ES6/7, TypeScript, Angular, React and Vue.js
- 12+ years IT trainer
- Voxxed Days, jPrime, jProfessionals, BGOUG, BGJUG, DEV.BG speaker
- Organizer RoboLearn hackathons and IoT enthusiast

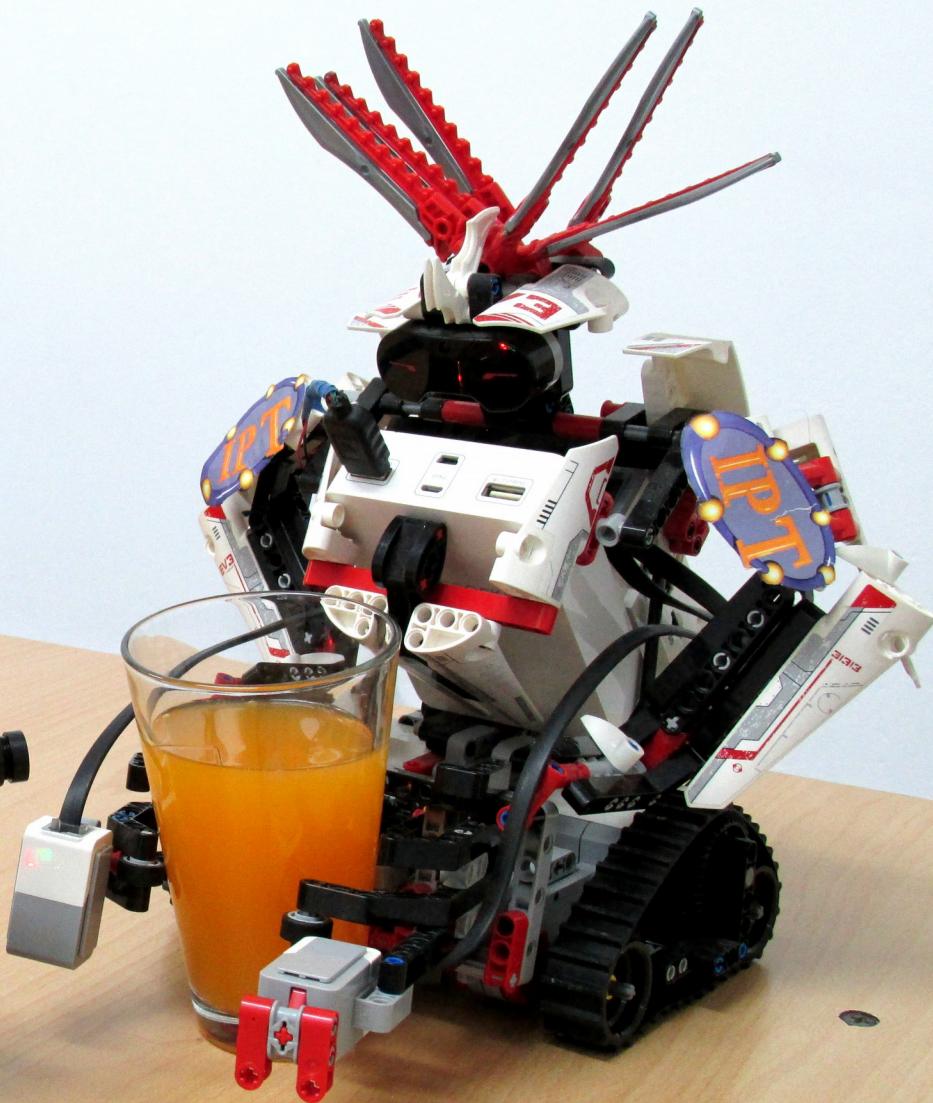
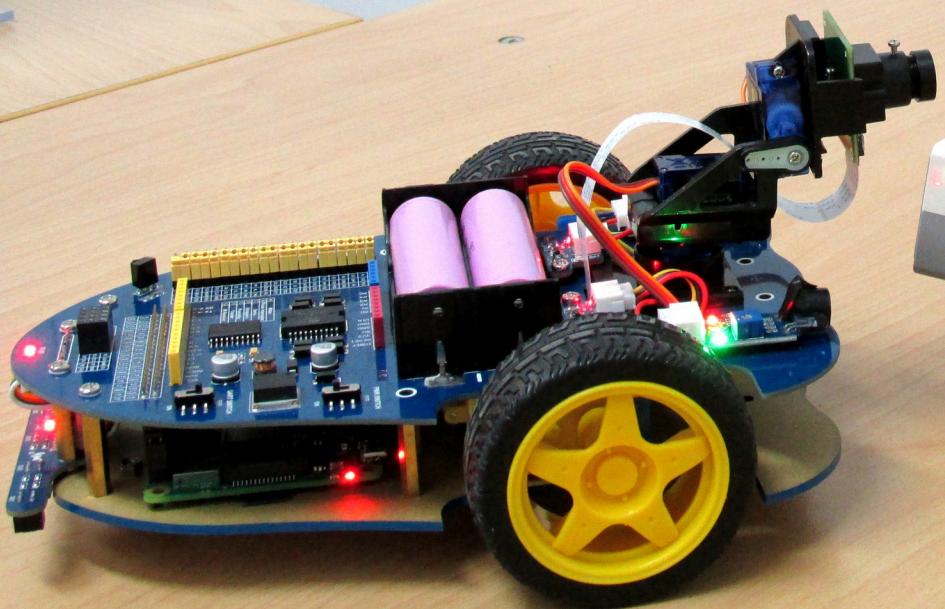
Trademarks

Oracle®, Java™ and JavaScript™ are trademarks or registered trademarks of Oracle and/or its affiliates.

LEGO® is a registered trademark of LEGO® Group. Programs are not affiliated, sponsored or endorsed by LEGO® Education or LEGO® Group.

Raspberry Pi™ is a trademark of Raspberry Pi Foundation.

Other names may be trademarks of their respective owners.



Where are The Code and Resources?

Course Practical Robotics and IoT in GitHub:

<https://github.com/iproduct/course-social-robotics/wiki/Study-Materials>

Why C++?

- C++ is very popular in Robotics and IoT – e.g. **Robot Operating System (ROS)** was written primarily in C++
- Highly efficient - high speed and low memory
- Optimal for real time event handling and control – used for programming micro-controllers
- Catching a lot of errors during compilation time
- C++ has an infinite number of libraries
- Naturally supported by **Arduino** open platform

Why Python? (1)

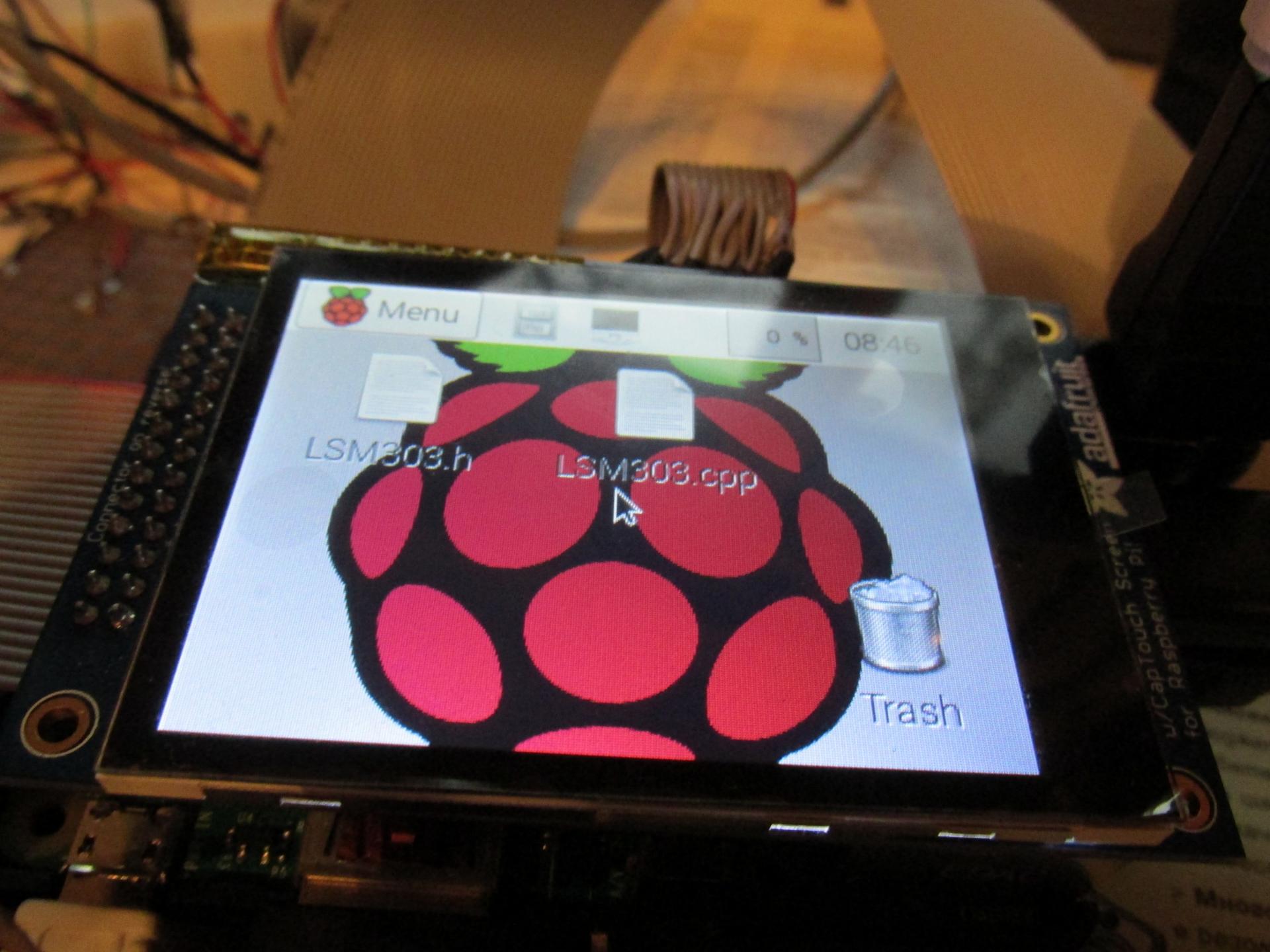
- Easy to learn scripting language – flat learning curve
- Very popular in **cloud computing, data science and machine learning** communities
- Good choice for Robotics and IoT learning – many examples – e.g. for **AlphaBot** robot platform
- You can make really short programs for complex things
- The final code of your node is quite easy to read and understand what it does.

Why Python? (2)

- You can do anything with Python. Python is a very powerful language with libraries for anything you want.
- It is easier to integrate it with web services based on **Flask** or **Django**. Since Django is based on Python, you can integrate ROS functions easily in the server calls.
- Major support in **ROS**
- Running event on constrained Arduino devices - e.g. **MicroPython** on the **ESP32** IoT platform

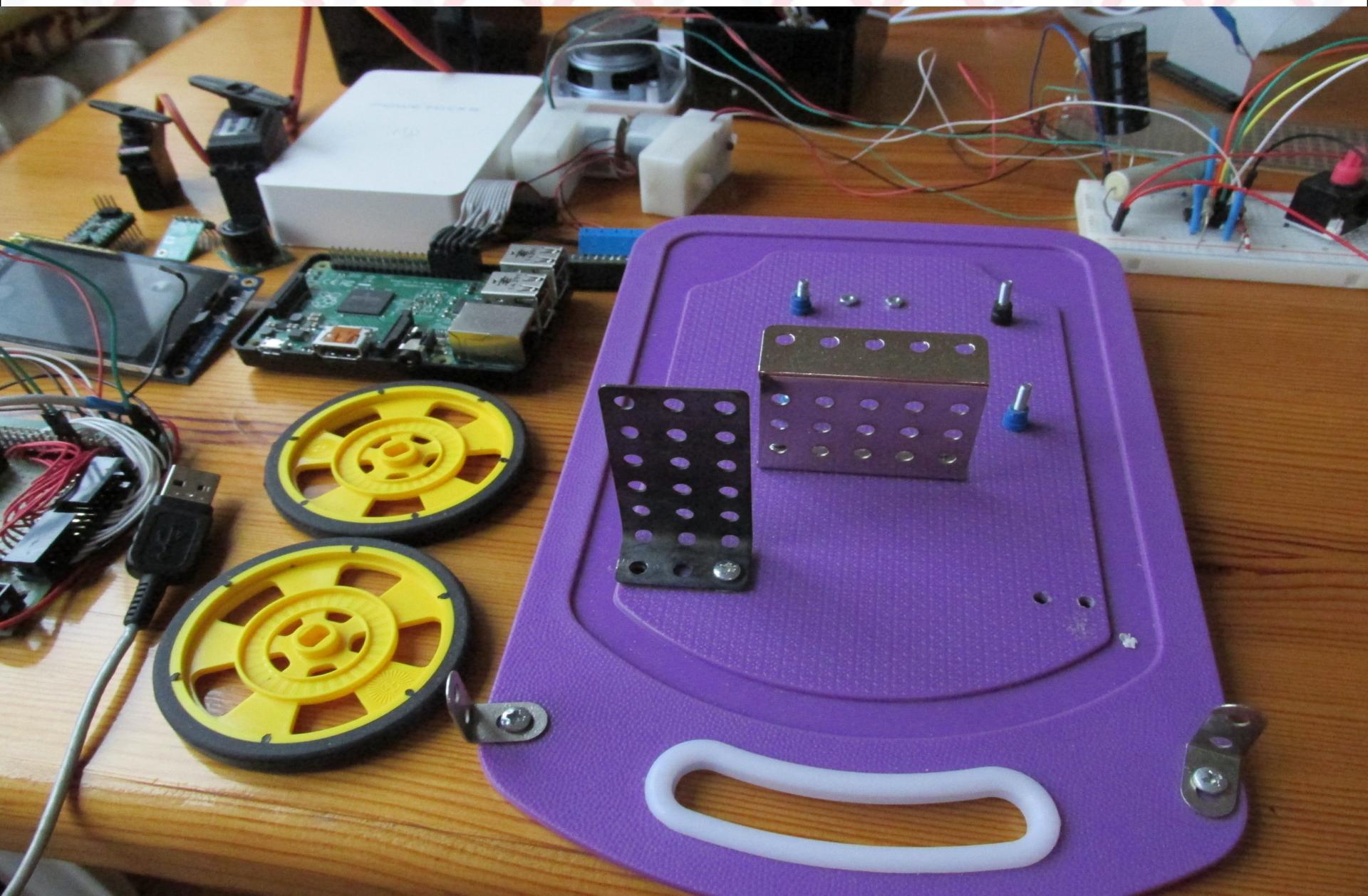
Why Java?

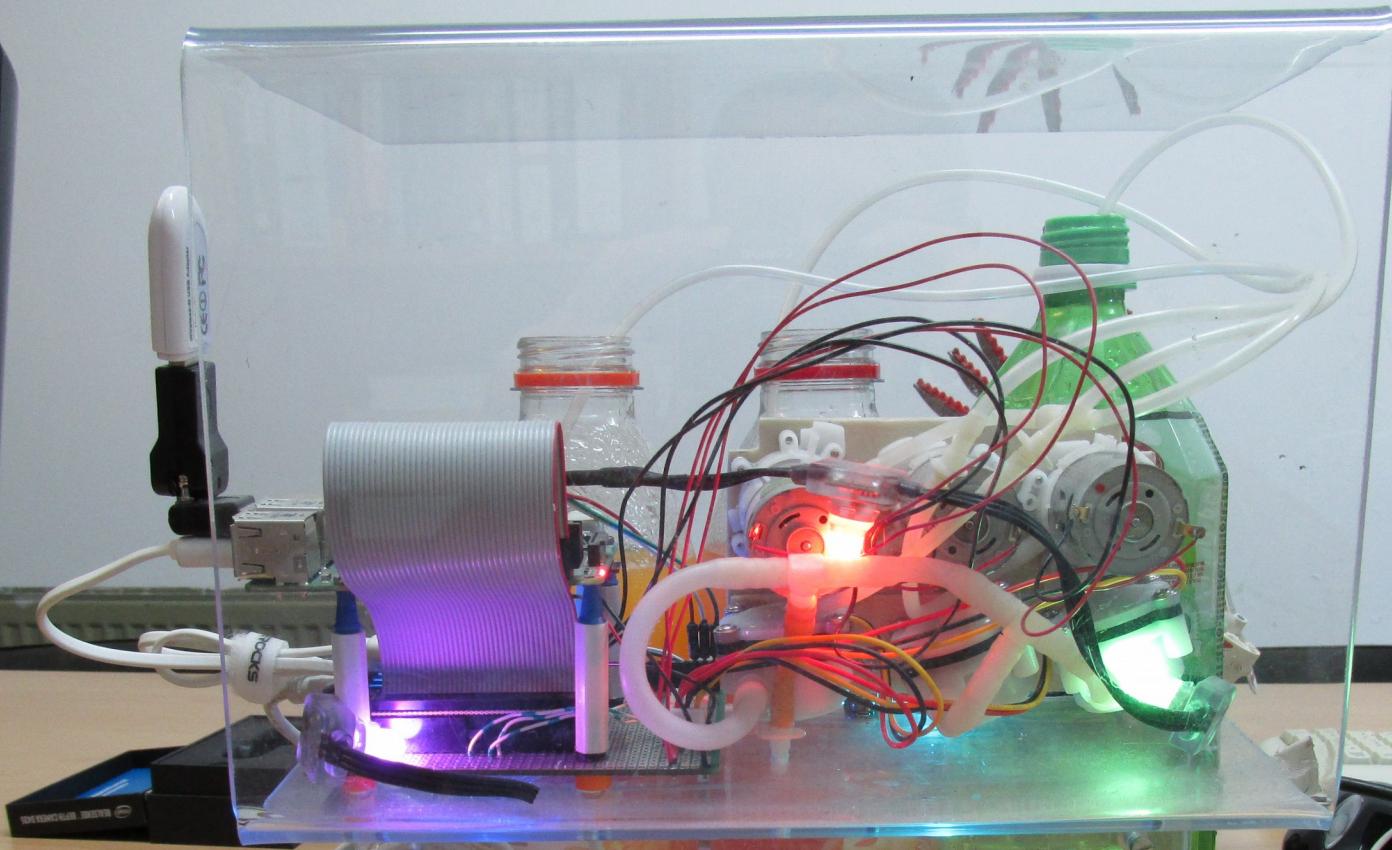
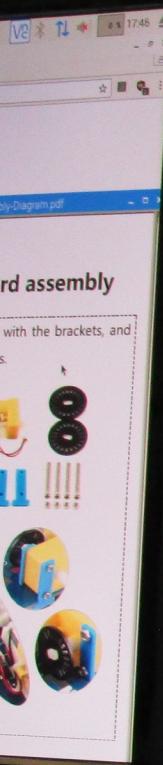
- A major OO programming language extensively used in the industry
- Running on **Raspberry Pi** – GPIO access using **Pi4J** lib
- Ideal for concurrent programming and **multi-threading**
- Java 8 Lambdas and Streaming
- Event driven and asynchronous programming
- Reactive programming – **Spring 5 WebFlux**, **Reactor**
- Serial and network communication (**USB**, **Web Sockets** support)
- Many, many libraries and integrations – **RabbitMQ**, **Kafka**, **Spark** (Distributed machine learning)

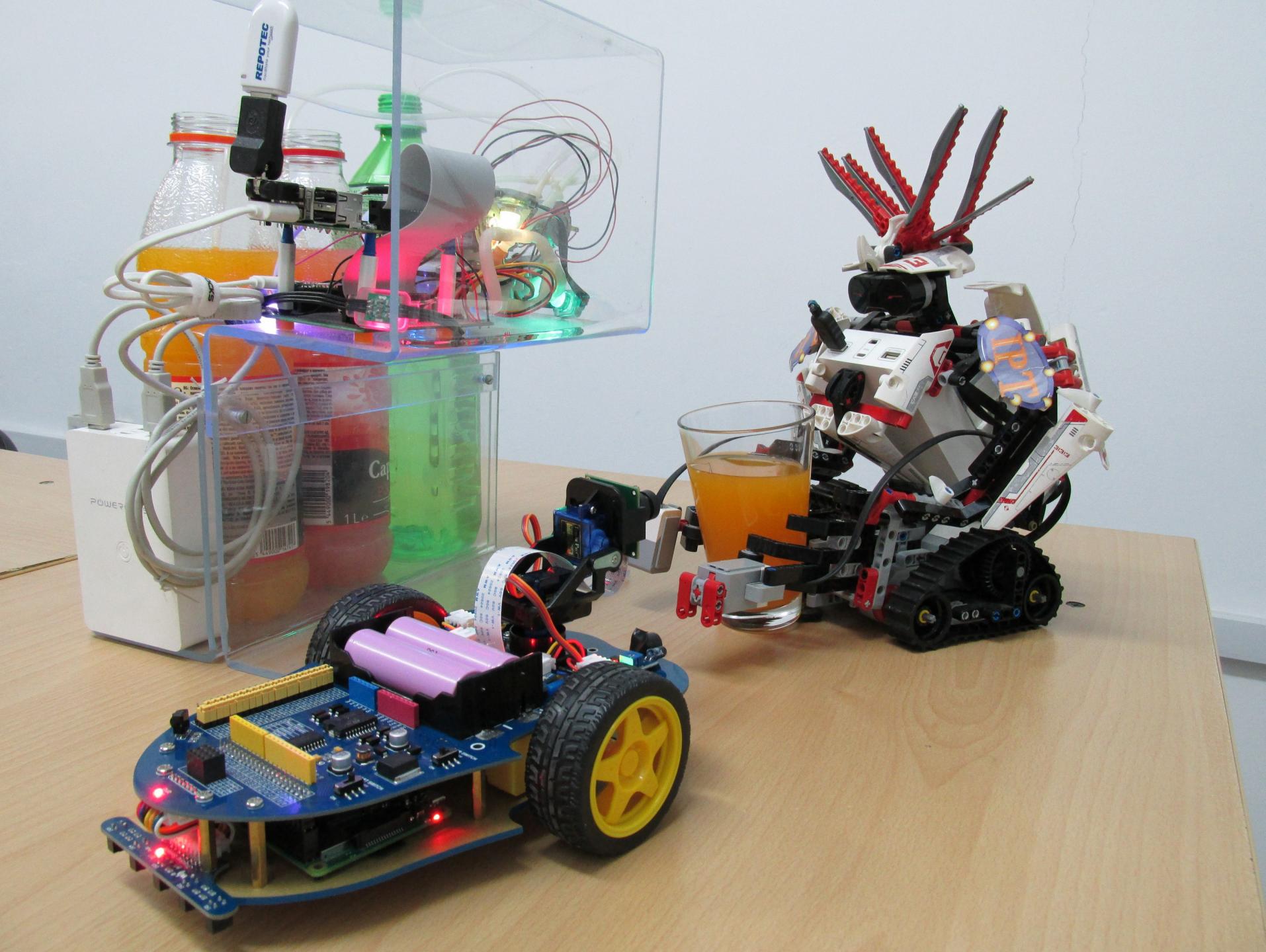


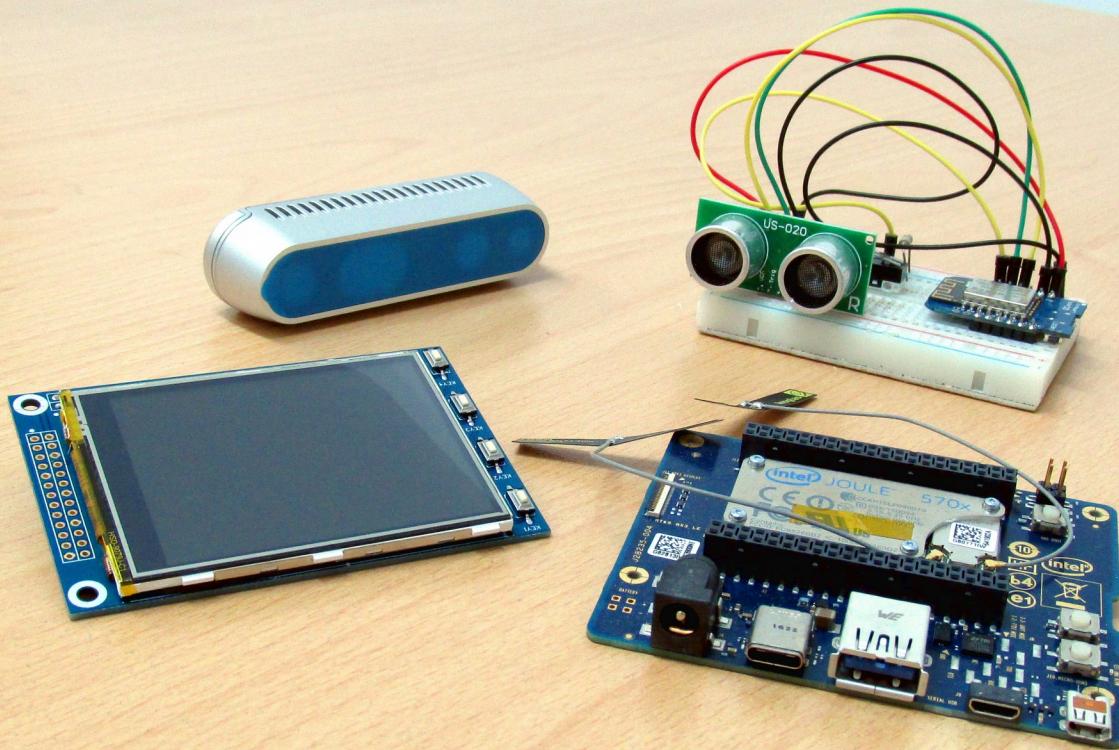
adafruit
Screen
CapTouch
for
Raspberry Pi

How to Build a Robot?









EEG-SMT

<http://openEEG.sourceforge.net>

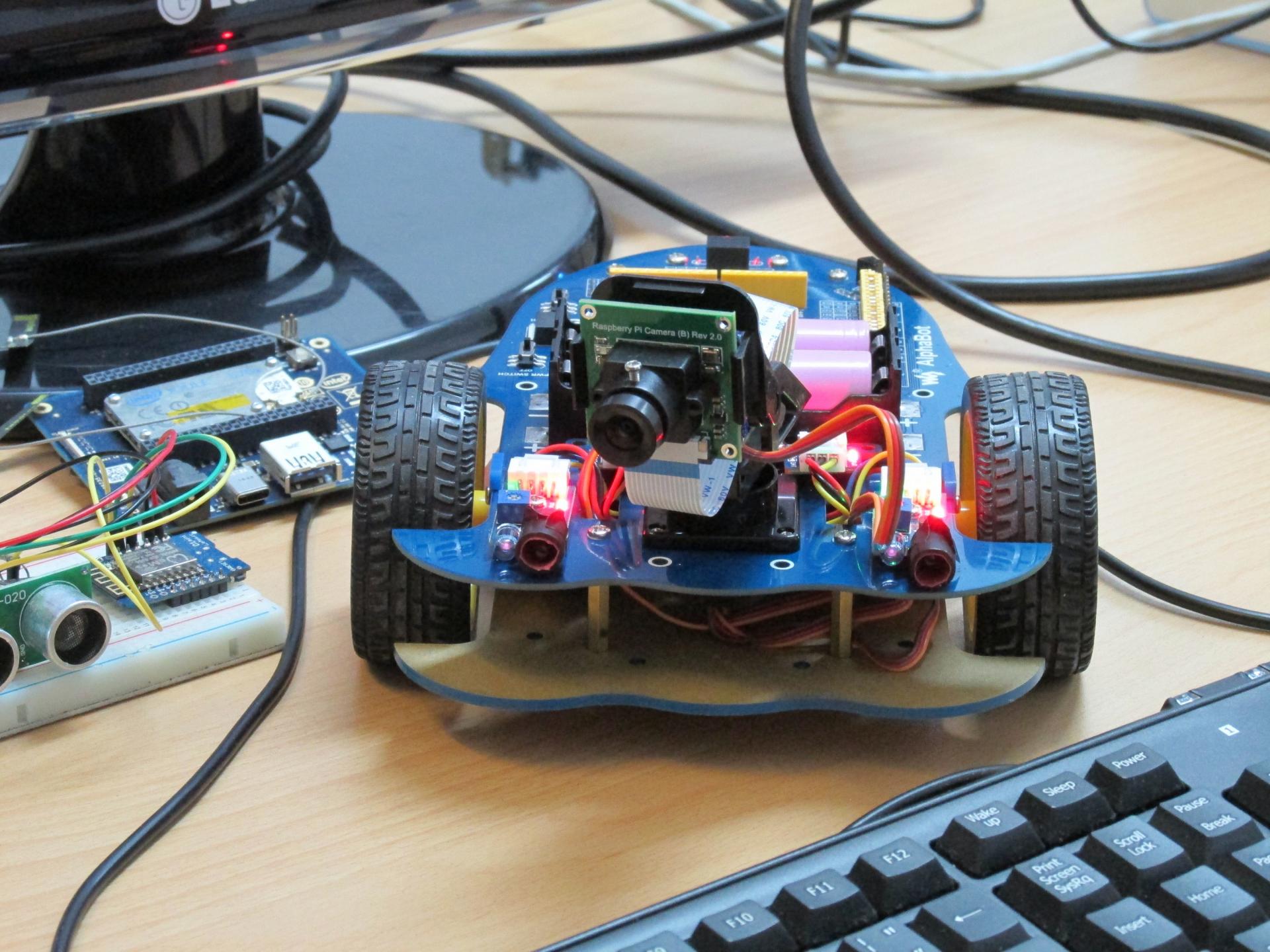
OLIMEX

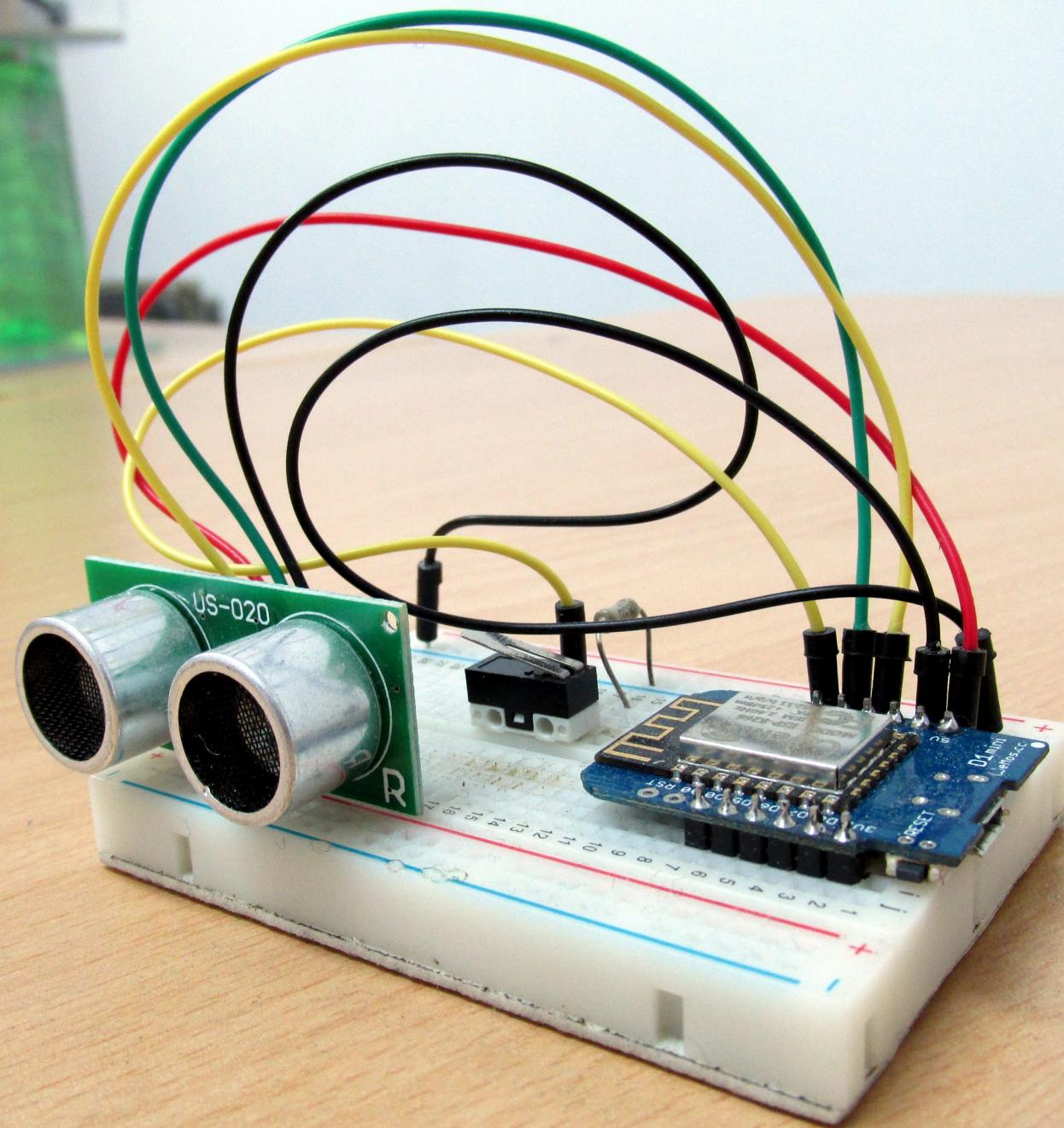
www.olimex.com

RMEPP30 - 21804

WÄHRINGEN
IE BROKEN

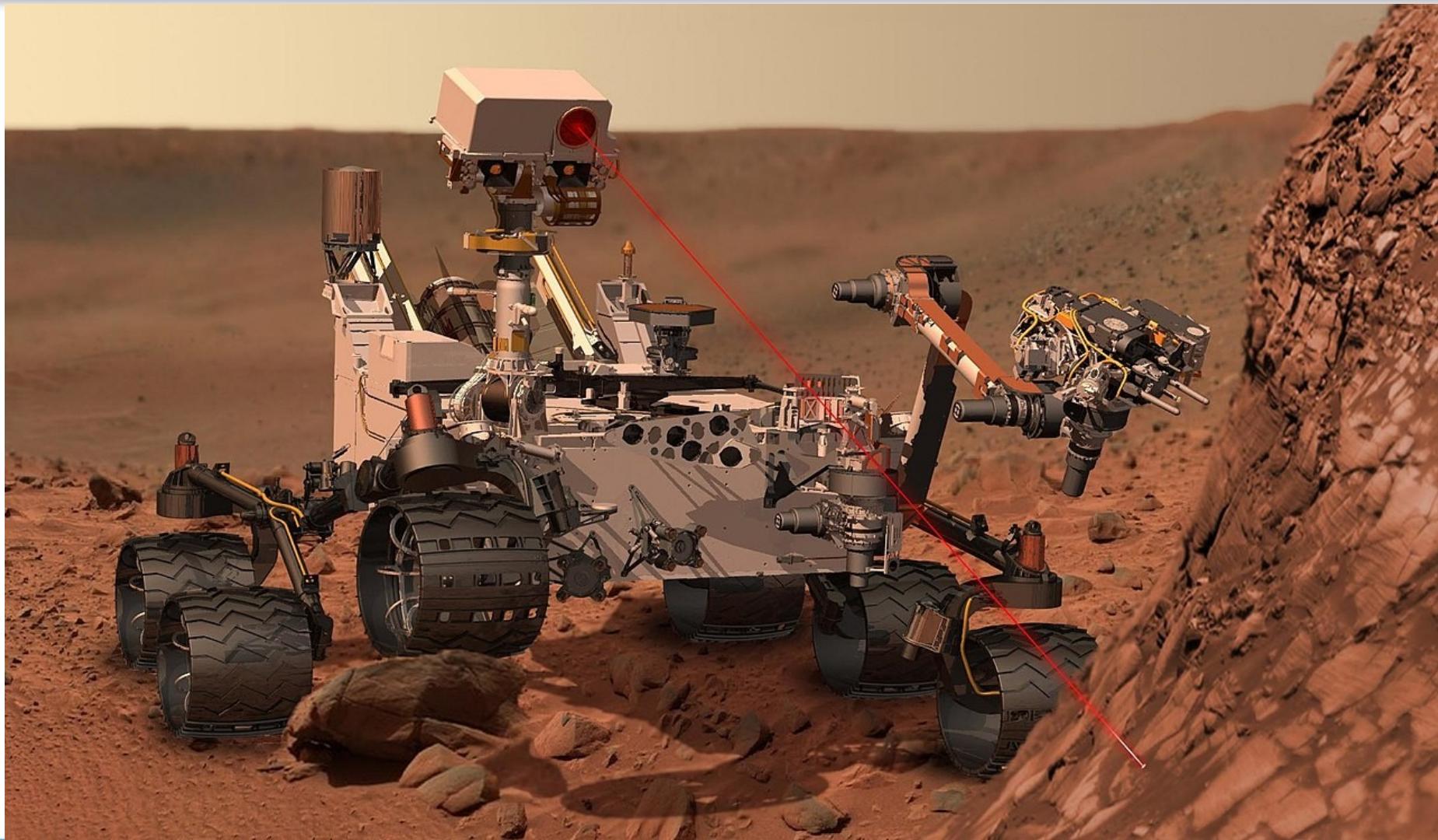
OLIMEX



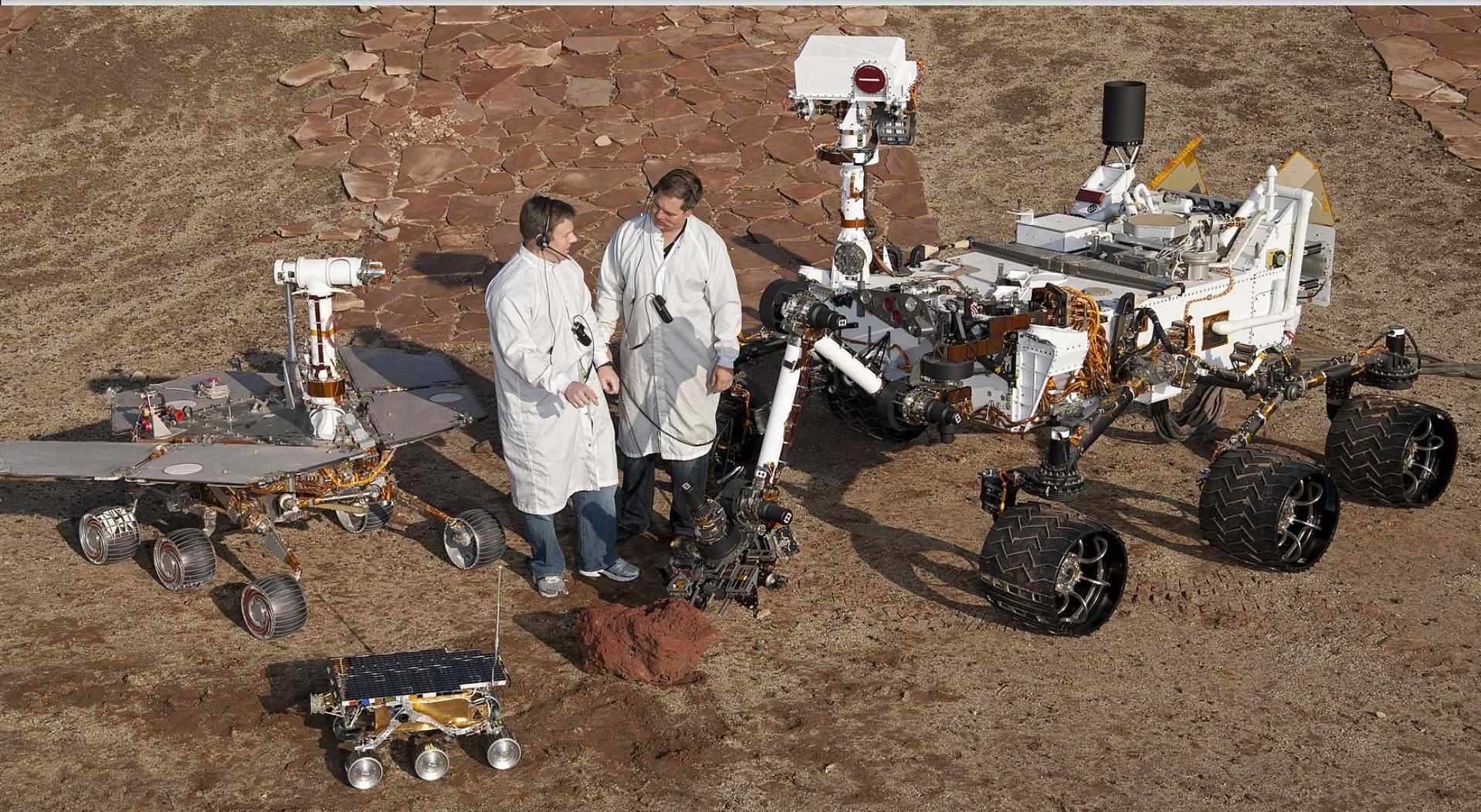




Where no man has been before...



With Mission to Explore...



Lonely Riders ...



By Arroww at the English language Wikipedia, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=1822443>

Удобно облекло :)



By DARPA - This file was derived from: DARPA Strategic Plan (2007).pdf, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=20798337>

Работите са вече у дома ...



Source: By Nohau - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=17987534>

... и как бихме пропуснали



Source: <https://commons.wikimedia.org/w/index.php?curid=234900>, CC BY-SA 3.0



Source: Korea Institute of Industrial Technology, <http://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=102&oid=020&aid=0000371339>

Във всякакви размери ...



Wearable Electronics :)

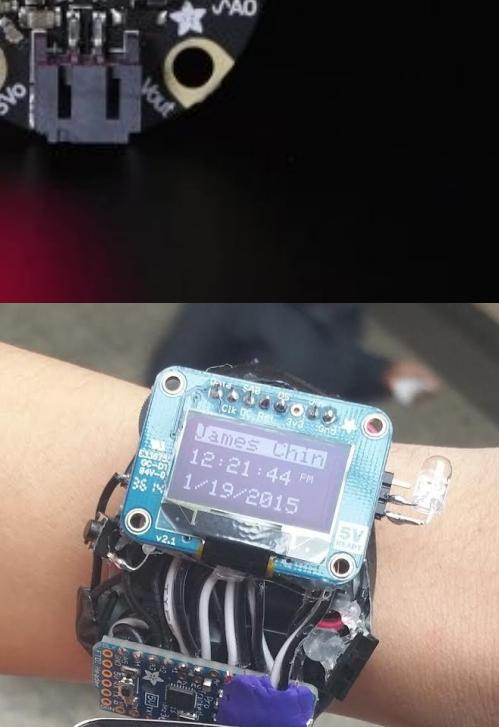
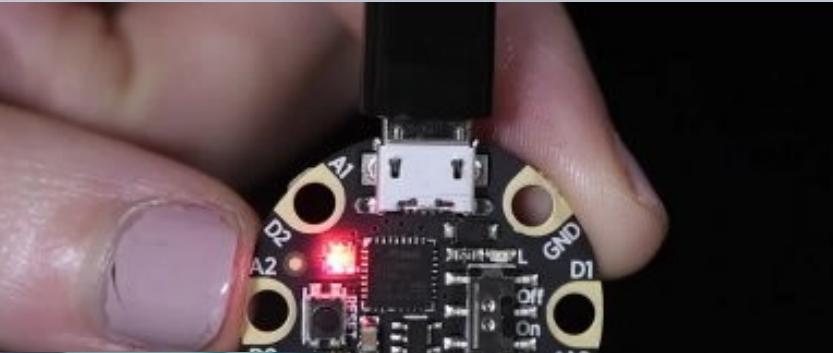


Photo credit: Adafruit, Becky Stern and others

Интернет на нещата (IoT, IoE)

Today computers—and, therefore, the Internet—are almost wholly dependent on human beings for information. Nearly all of the roughly 50 petabytes (a petabyte is 1,024 terabytes) of data available on the Internet were first **captured and created by human beings**—by typing, pressing a record button, taking a digital picture, or scanning a bar code. ... The problem is, people have limited time, attention and accuracy—all of which means they are not very good at capturing data about **things in the real world**. ... We're physical, and so is our environment ... If we had computers that knew everything there was to know about things ... we would be able to track and count everything, and **greatly reduce waste, loss and cost**. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best. **The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so.**

— Kevin Ashton, 'That 'Internet of Things' Thing', RFID Journal,
July 22, 2009



Интернет на нещата (IoT, IoE)

Интернет на нещата има потенциала да промени света, така както го направи Интернет. Дори повече.

— Kevin Ashton, 'That 'Internet of Things' Thing', RFID Journal, 2009

- ❖ 50 петабайта данни се **създават и въвеждат от хора**
- ❖ Хората имат ограничено време, внимание и точност
- ❖ Данни за **нещата от реалния свят в реално време**
- ❖ Проследяване и управление на всичко, **намаляване на проблемите, загубите и цената**
- ❖ Ще знаем кога нещата се нуждаят от **замяна, ремонт или извеждане от експлоатация**



Интернет на нещата в цифри

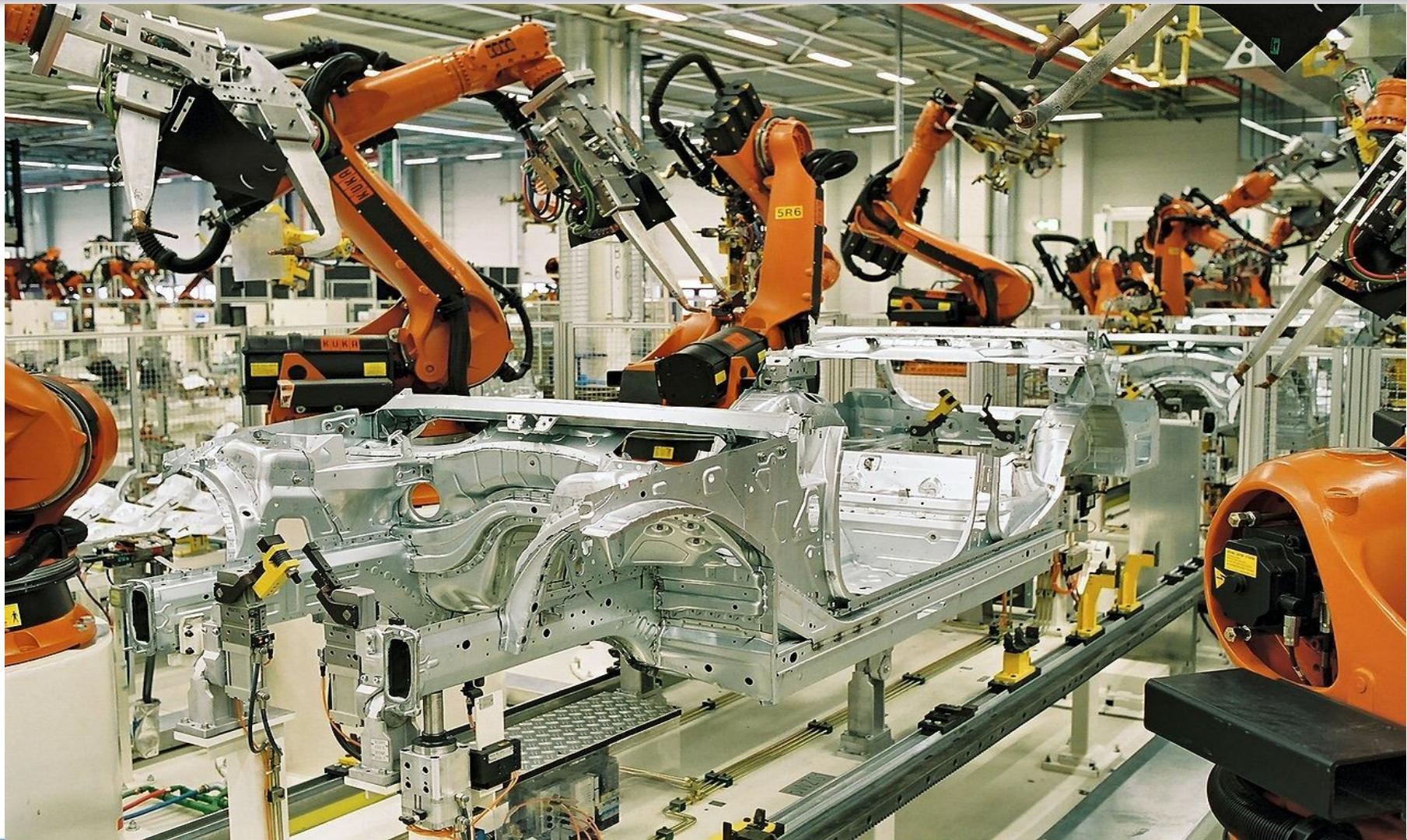
- According to Gartner, there will be nearly **26 billion** devices on the Internet of Things by 2020.
[Gartner, 2013-12-12,
<http://www.gartner.com/newsroom/id/2636073>]
- According to ABI Research, more than **30 billion** devices will be wirelessly connected to the Internet of Things by 2020 (Internet of Everything)
[ABI Research, <https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne>]
- It's expected to be a **19 Trillion USD** market
[John Chambers, Cisco CEO,
<http://www.bloomberg.com/news/2014-01-08/cisco-ceo-pegs-internet-of-things-as-19-trillion-market.html>]

Internet of Things & AI

“Artificial intelligence (AI) is viewed as vital to realising the value of IoT data. Just over a quarter of survey respondents (26%) say that IoT data are pivotal to their current or planned use of AI, with 56% identifying IoT as “one of many important sources” for AI initiatives. Furthermore, 64% agree that “the value of IoT data to my organisation has increased as we have developed our AI capabilities”. Many interviewees view IoT and AI as two components of an advanced analytics capability. Reportedly, algorithms trained on data sources including IoT provide the greatest value and competitive differentiation.”

[The Economist Intelligence Unit Limited 2020:
<https://learn.arm.com/economist-iot-report-typ.html>]

В промышленности ...



Internet of Things (IoT) – архитектурни изисквания

"Basket of remotes" problem – we'll have hundreds of applications to interface with hundreds of devices that **don't share protocols for speaking with one another**

[Jean-Louis Gassée, Apple initial alumni team, and BeOS co-founder,
<http://www.mondaynote.com/2014/01/12/internet-of-things-the-basket-of-remotes-problem/>]

- IoT устройствата трябва да бъдат лесно достъпни за своите потребители и техните агенти
- IoT архитектурите трябва да бъдат конкурентни разпределени, устойчиви на грешки, високо-производителни, еластични и децентрализирани, **динамично** разширяеми и еволюиращи

Работите са интелигентни “неша”



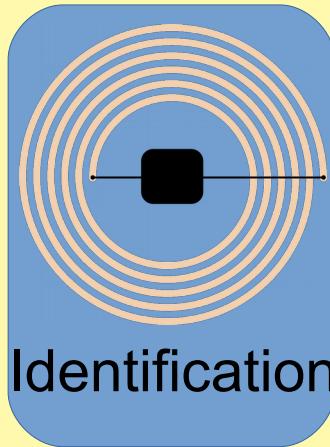
Radar, GPS, lidar for navigation and obstacle avoidance (2007 DARPA Urban Challenge)

CC BY 2.0, Source:
<https://www.flickr.com/photos/wilgengebroed/8249565455/>



Key Elements of IoT

Internet of Things (IoT)



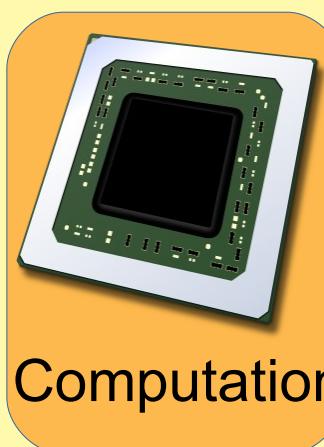
Identification



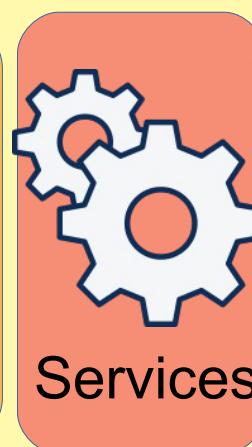
Sensors



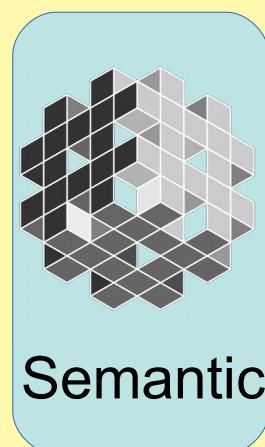
Connectivity



Computation

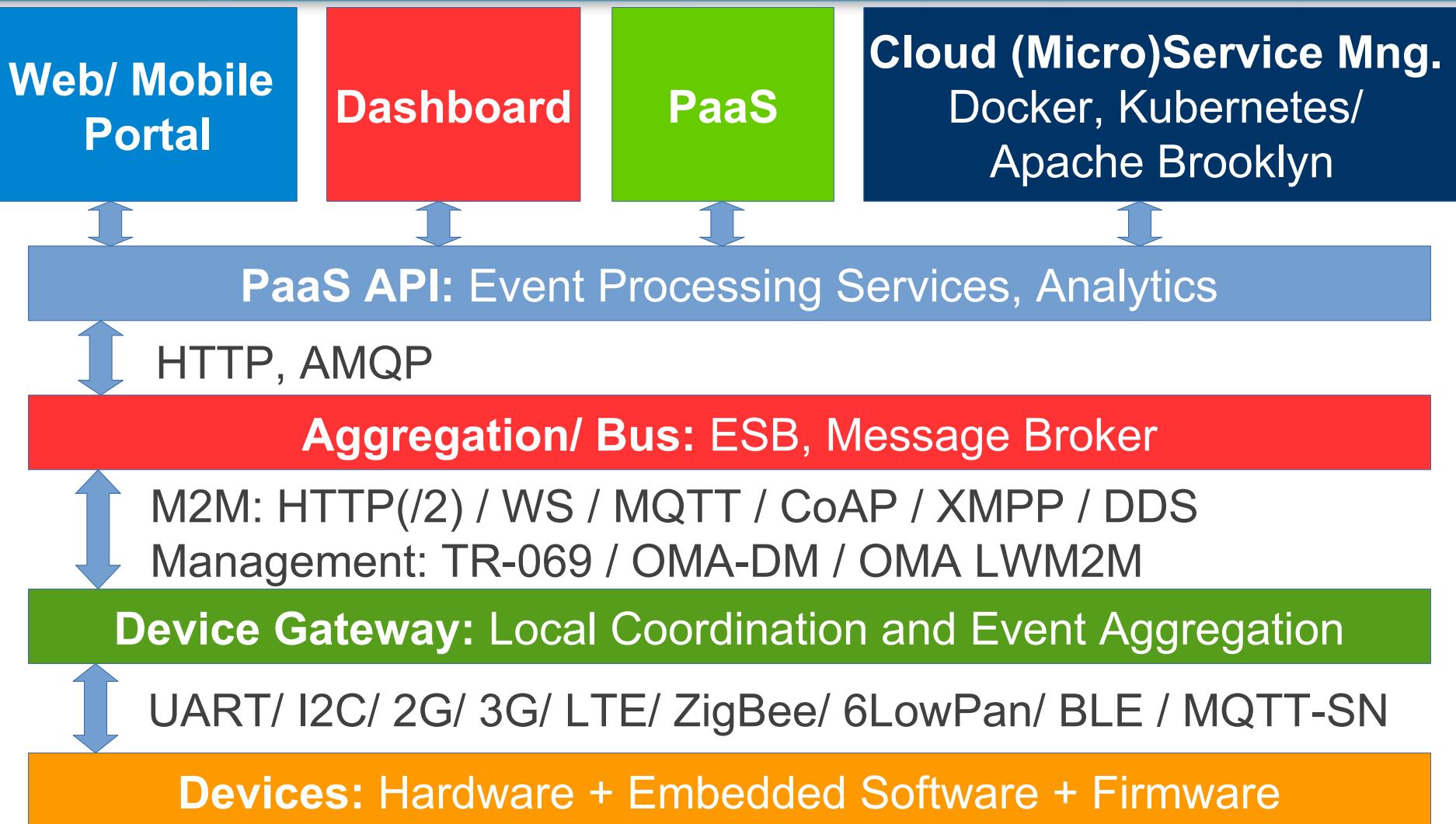


Services



Semantic

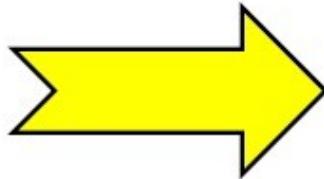
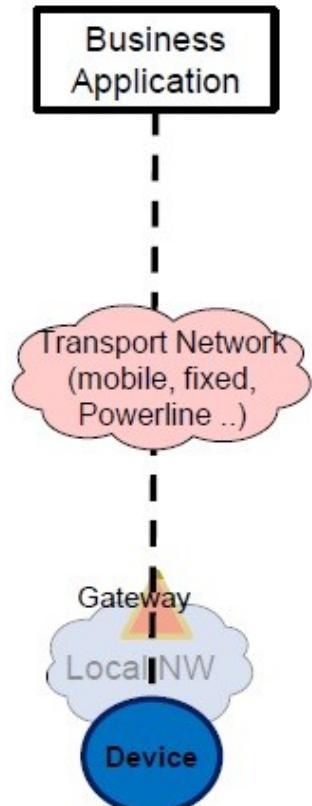
IoT Services Architecture



Vertical vs. Horizontal IoT

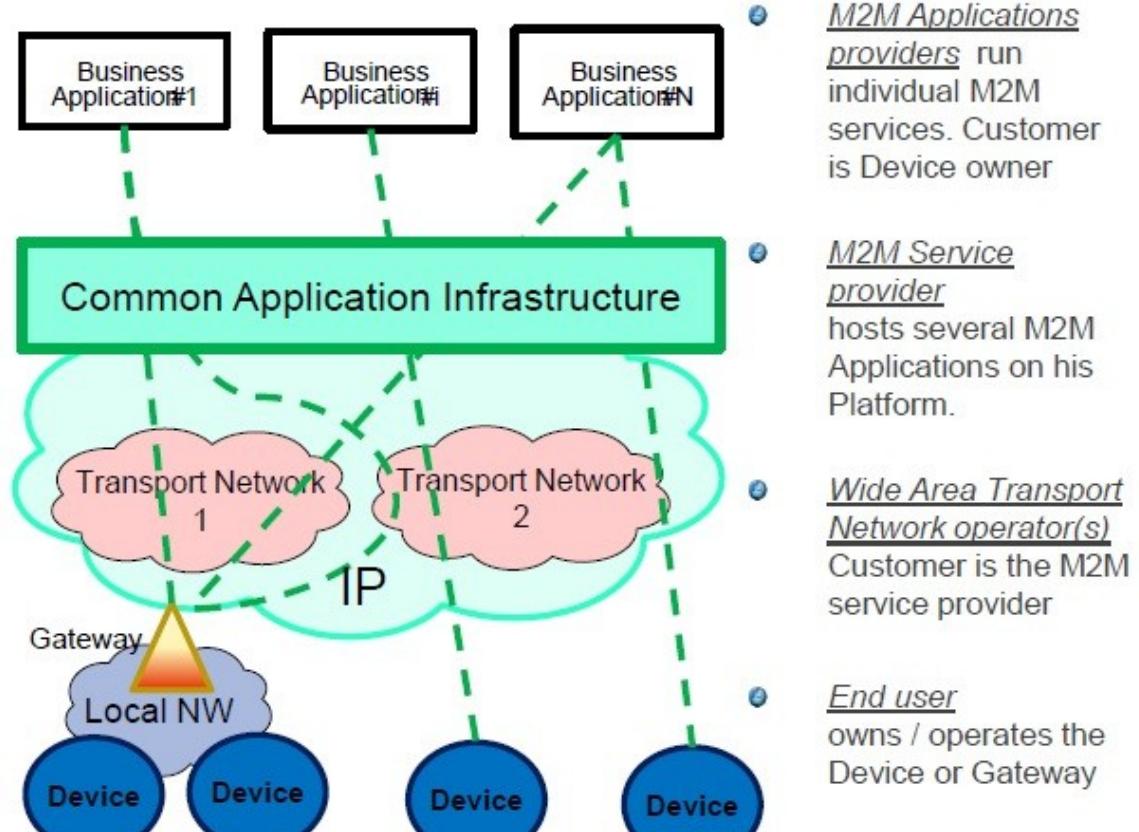
Pipe (vertical):

1 Application, 1 NW,
1 (or few) type of Device



Horizontal (based on common Layer)

Applications share common infrastructure, environments
and network elements



Cloud, Fog and Mist Computing

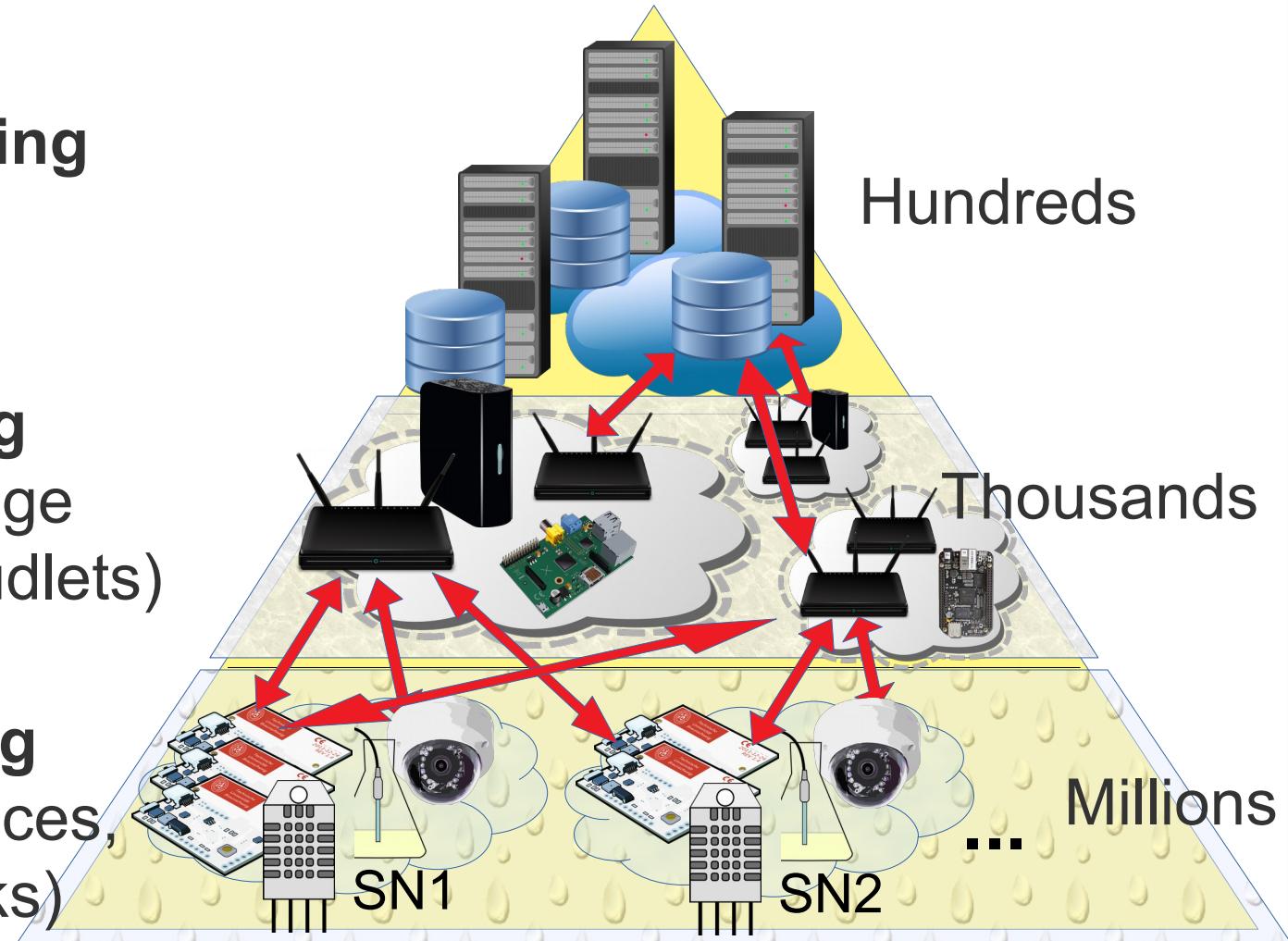
Cloud Computing
(Data-centers)



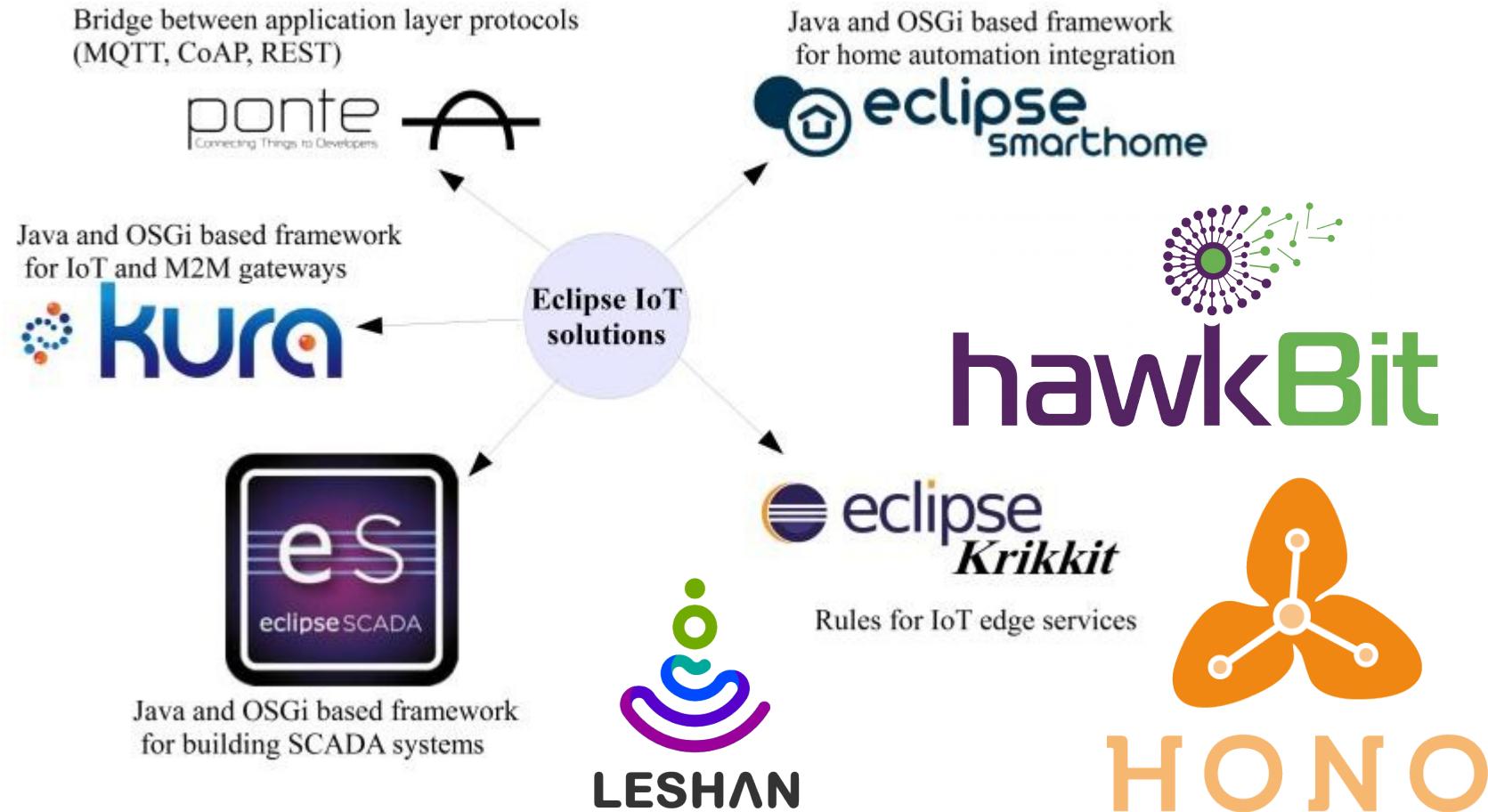
Fog Computing
(Fog Nodes, Edge
Gateways, Cloudlets)



Mist Computing
(Smart IoT Devices,
Sensor Networks)



Eclipse IoT Platform



Based on: https://www.researchgate.net/publication/279177017_Internet_of_Things_A_Survey_on_Enabling_Technologies_Proocols_and_Applications, By Ala Al-Fuqaha et al. - Internet of Things: A Survey on Enabling Technologies, Protocols and Applications

Интер-дисциплинарно обучение

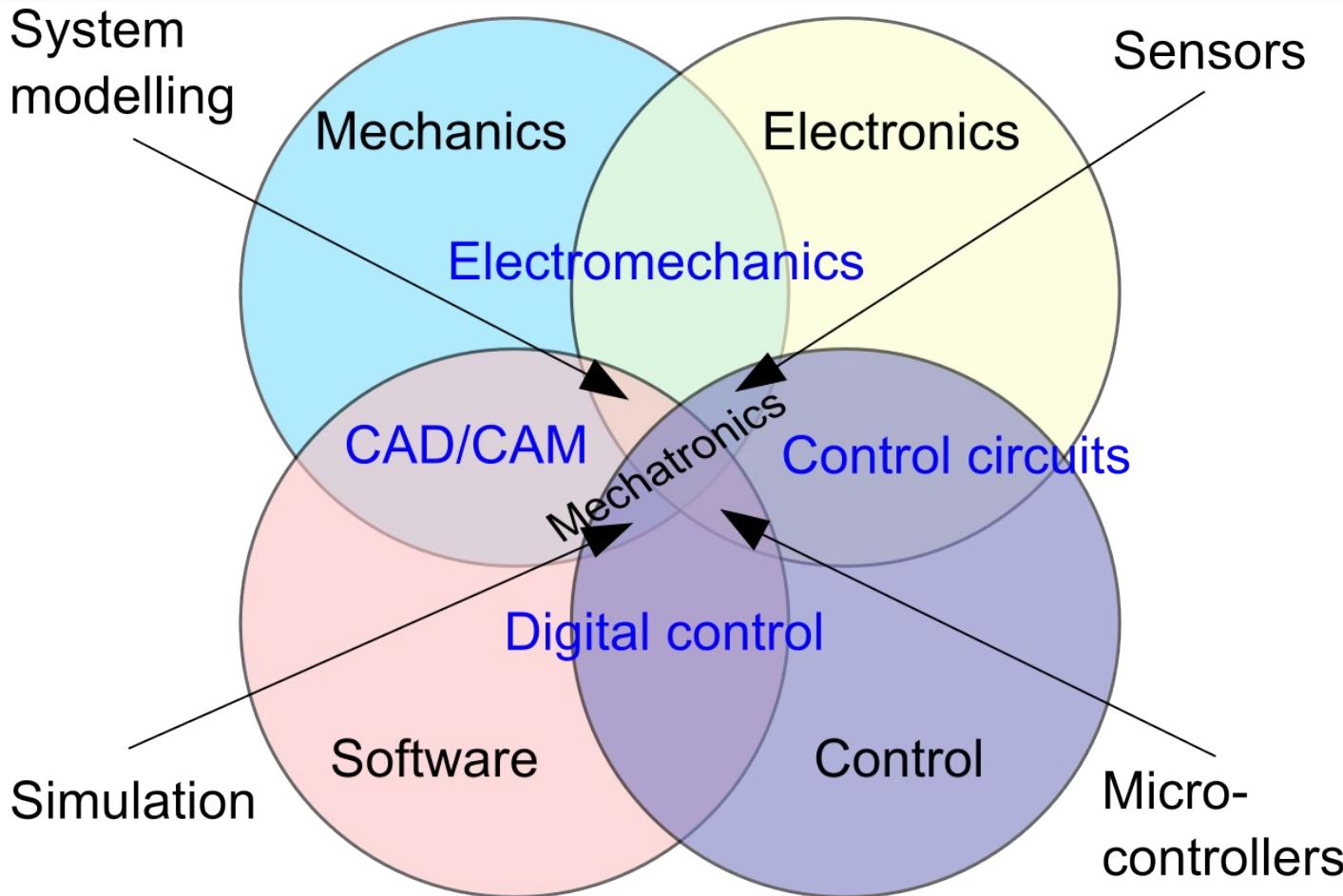


Роботиката е пресечна точка на
много дисциплини:

- ❖ Механика
- ❖ Електроника и компютърни науки
- ❖ Софтуерно инженерство
- ❖ Изкуствен интелект (AI)
- ❖ Човеко-машинни интерфейси
- ❖ Социология и психология
- ❖ Дизайн

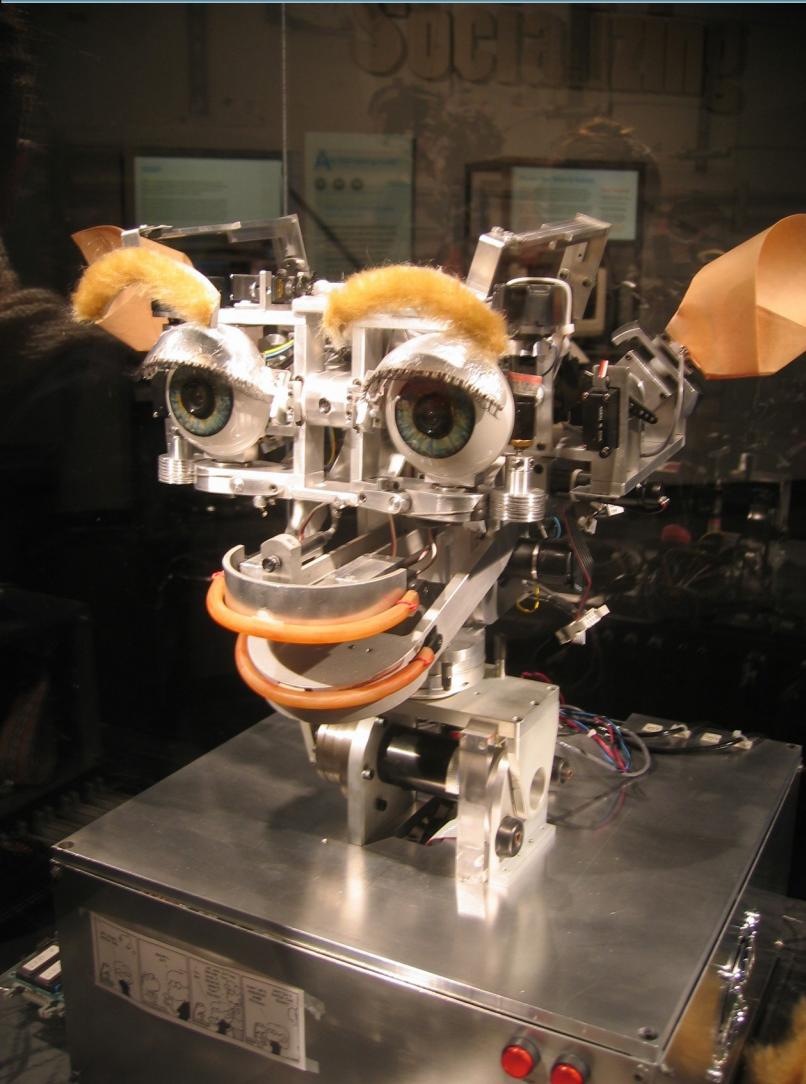
Picture by Hugo Elias of the Shadow Robot Company -
<http://www.shadowrobot.com/media/pictures.shtml>, CC BY-SA 3.0

Инженерство, наука и искусство



Source: <https://commons.wikimedia.org/w/index.php?curid=551256>, CC BY-SA 3.0

Роботите могат да бъдат социални :)



Обучение чрез програмиране на малки роботи





Documentation FAQ Download Mailing List Code Commercial Support

Build powerful concurrent & distributed applications more easily.

Akka is a toolkit and runtime for building highly concurrent, distributed, and resilient message-driven applications on the JVM.

Simple Concurrency & Distribution

Asynchronous and Distributed by design. High-level abstractions like Actors, Futures and STM.

Resilient by Design

Write systems that self-heal. Remote and/or local supervisor hierarchies.



High Performance

50 million msg/sec on a single machine. Small memory footprint; ~2.5 million actors per GB of heap.

Elastic & Decentralized

Adaptive load balancing, routing, partitioning and configuration-driven remoting.

Extensible

Use Akka Extensions to adapt Akka to fit your needs.

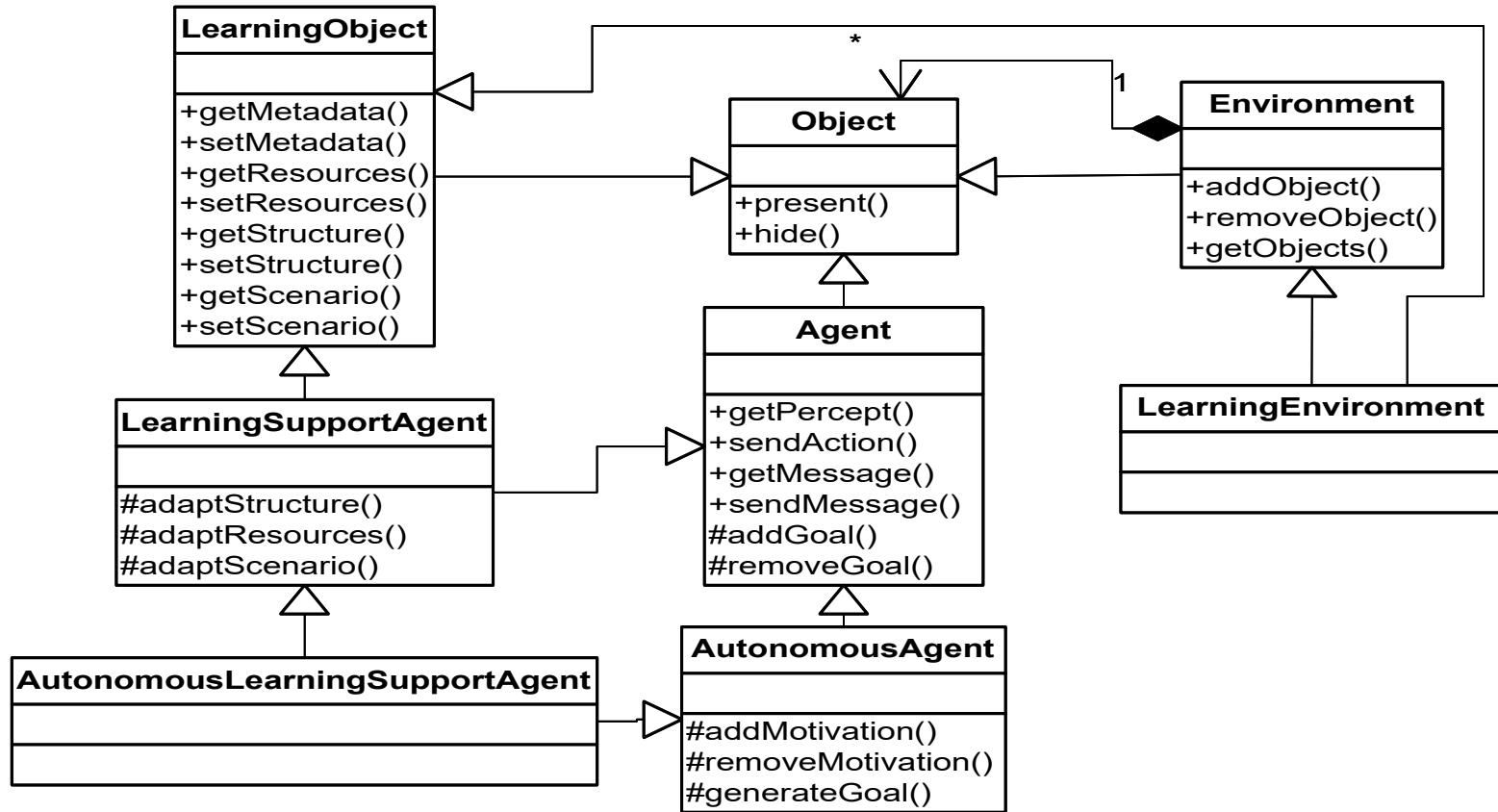
Подход на интелигентните агенти при моделиране на знания и системи

- Интелигентните агенти (ИА) са софтуерни/хардуерни системи, които притежават свойствата:
 - Автономност
 - Способност за реагиране (реактивност)
 - Проактивност
 - Социална способност (комуникация, ACL)
 - Рационалност
 - Адаптивност (Самообучение)
 - Мобилност
- ИА възприемат средата с помощта на сензори и ѝ въздействат с помощта на ефектори

Дефиниции за интелигентни агенти

- Интелигентните технологии в нашия живот – визия за бъдещето
- Дефиниции:
 - Intelligent Agent = Agent + Intelligence
 - AIMA agent (Russel & Norvig, 1995)
 - Maes agent (Maes, 1995)
 - Hayes-Roth agent (Hayes-Roth, 1995)
 - SodaBot агенти
 - Foner агенти
 - Wooldridge и Jennings агенти
 - Franklin и Graesser агенти
- Формализация на Luck и d'Inverno на понятията свързани с агентите

Пример: Learning Support Agents [Iliev, ICL2003]



Характеристики на интелигентните агенти

- Автономност
- Способност за реагиране (реактивност)
- Проактивност
- Социална способност (комуникация, ACL)
- Рационалност
- Доброжелателност и честност
- Времева продължителност
- Адаптивност (Самообучение)
- Личност и антропо-морфизъм
- Мобилност

Автономност на интелигентните агенти

- “Автономността е трудно за прецизно дефиниране понятие и ние го разглеждаме в смисъл, че системата трябва да може да действа без директна интервенция от страна на хора (или други агенти) и че тя трябва да притежава контрол върху своите собствени действия и вътрешно състояние“

[Jennings, Sycara и Wooldridge]

- “Една система е автономна в степента, в която нейното поведение се определя от нейния собствен опит“

[Russell и Norvig, AIMA]

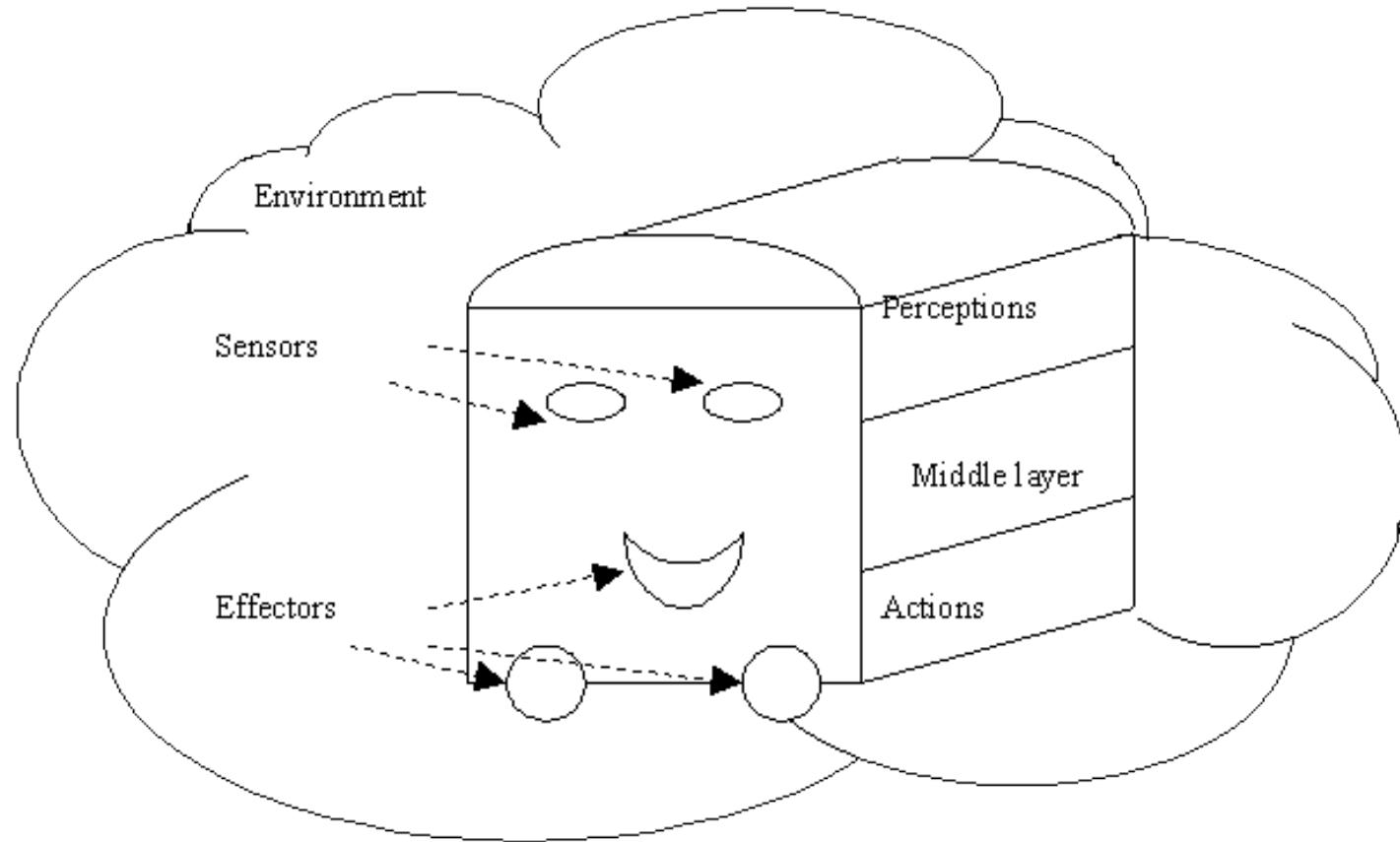
Автономност на интелигентните агенти 2

- Luck и d'Inverno – две възможни гледни точки към автономността:
 - **силна гледна точка към автономността** – “разглеждана като абсолютна без измерения или мярка за степен”;
 - **слаба гледна точка към автономността** – практически ориентирана, “при която автономността се счита за синоним на независимостта, едно явно относително понятие”.

Видове агенти

- Сътрудничещи си (Collaborative)
- Интерфейсни (Interface)
- Информационни (Information)
- Реагиращи (Reactive)
- Мобилни (Mobile)
- Хибридни (Hybrid)
- Хетерогенни (Heterogeneous)
- **Типове приложения**

Базова архитектура на интелигентен агент



Intelligent agent – basic architecture

Основни парадигми при МАС и ИИ роботиката

- **Йерархична парадигма** – планиране, символно представяне на знанията и глобален модел на света. Три слоя: Sensing, Planning, Acting [Murphy].
- **Реактивна парадигма** – “intelligent behavior emerges from the interaction of various simpler behaviors” [Wooldridge]: Subsumption Architecture [Brooks].
- **Хибридна Deliberative/Reactive** парадигма – синтез на предишните два подхода: INTERRAP [Muler].

Стандарти при управлението на знания. Онтологии.

- Семантична мрежа - W3C Resources Description Framework (RDF)
- Онтологии – W3C Web Ontology Language (OWL):
 - Концептуализация
 - Речник
 - Аксиоматизация

Основни парадигми в ИИ роботиката

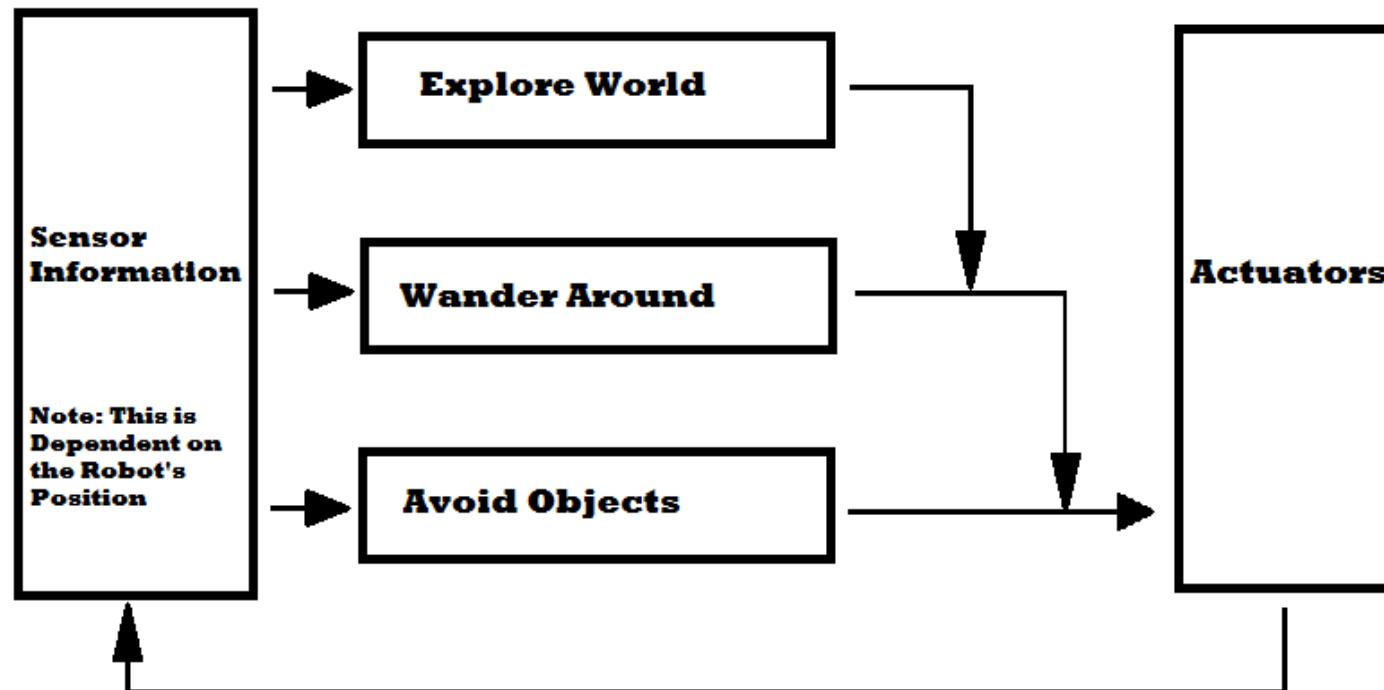
- Йерархична парадигма – подход на планиране [STRIPS, <http://en.wikipedia.org/wiki/STRIPS>]
- Реактивна парадигма [Brooks, Subsumption Architecture]
- Хибридна парадигма [INTERRAP]:
 - Реагиране
 - Планиране
 - Социална комуникация и взаимодействие

Subsumption Architecture [Brooks, 1986 & 1991]

- Rodney A. Brooks, MIT AI Lab — статия "Elephants Don't Play Chess", 1986
- Бихевиористки подход към реализацията на изкуствен интелект – **Behavior-Based Robotics (BBR)**.
- BBR роботите действат на база множество независими, прости поведения, които могат да се надграждат йерархично. Поведенията имат тригер (сензорна перцепция) и действие (включващо ефектор). Когато две поведения са в конфликт, той се решава от Арбитратор.
- Интелигентното поведение не е просто сума на отделните прости поведения, а е **възникващо (emergent)**.

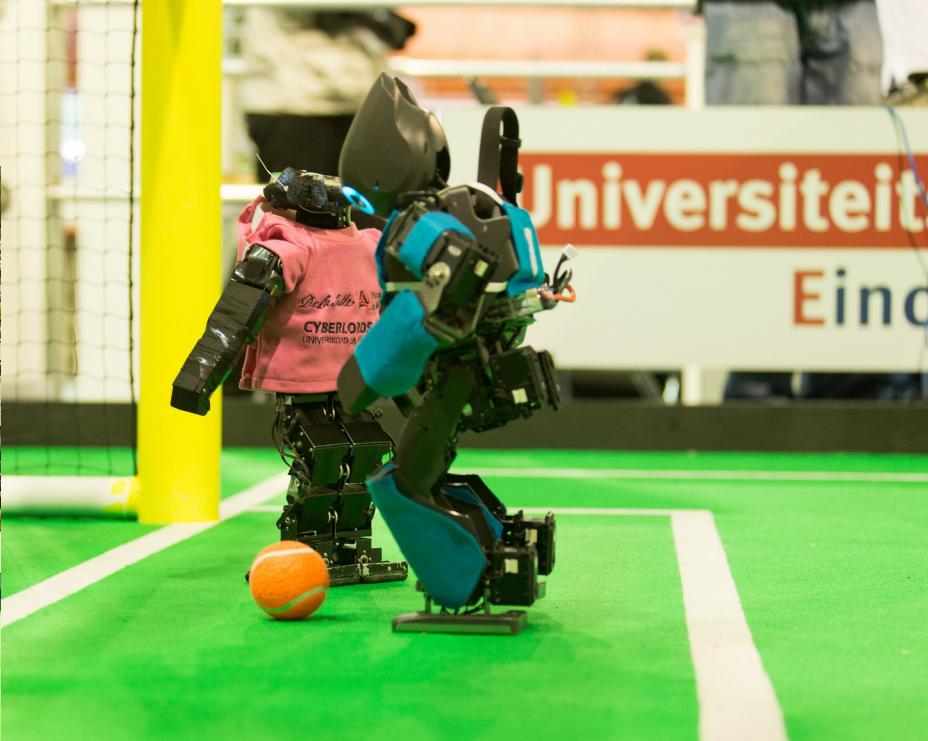
Subsumption Architecture [Brooks, 1986 &1991]

- Situatedness
- Embodiment
- Intelligence
- Emergence

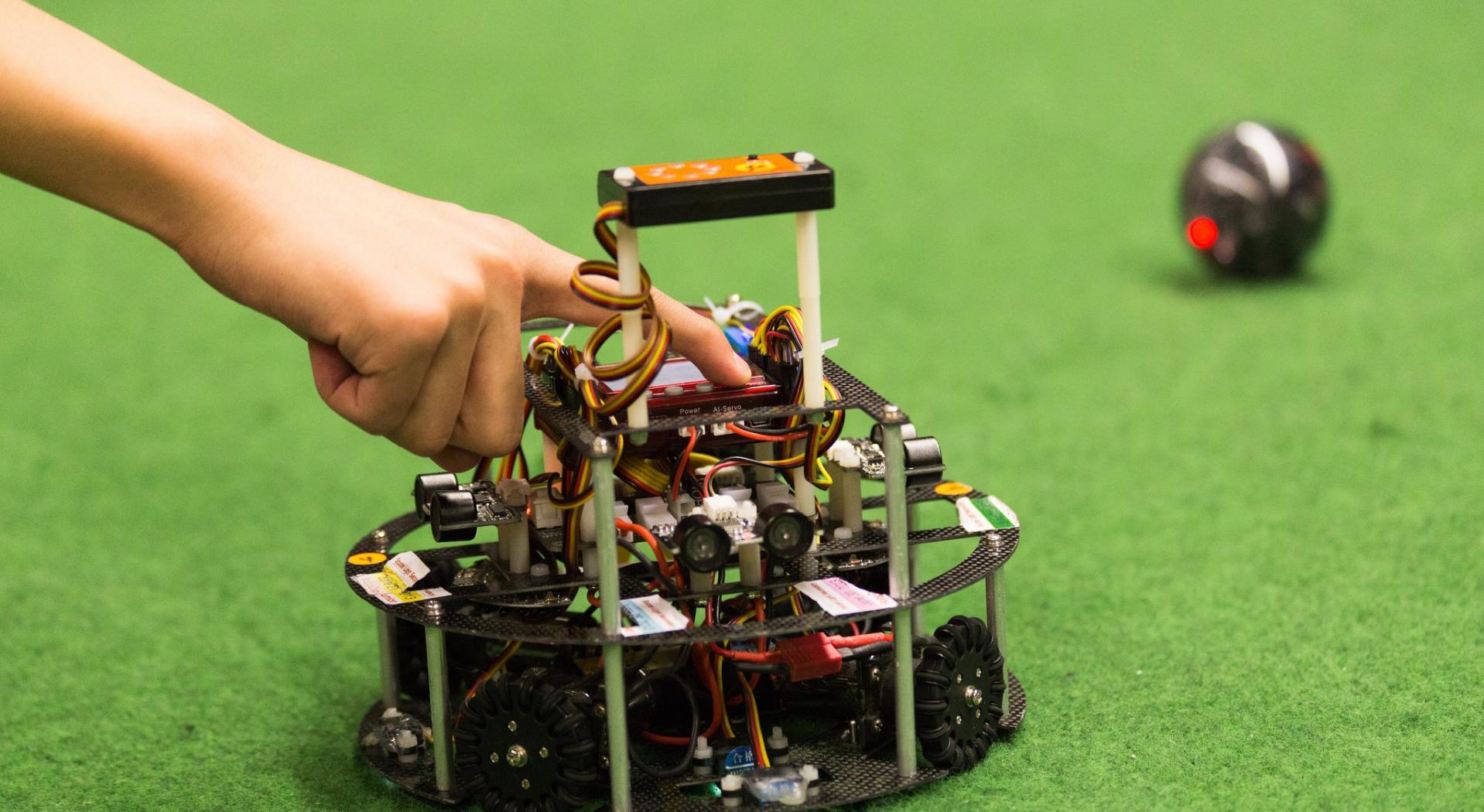


Много-агентни системи (MAS)

- Отделните агенти вземат решения и действат автономно, на базата на **собствен модел на света**
- Резултатът от взаимодействието се определя от съвместните действия на всички агенти => възникващо поведение на системата (*emergent behavior*)
- Необходимо е да се дефинира **механизъм**, по който става взаимодействието между агентите
- Различните агенти могат да имат **различни функции на полезност**, които им позволяват да вземат различни решения



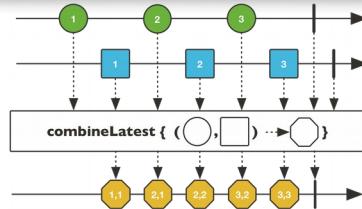
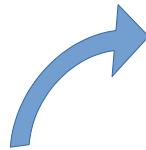
AvB - RoboCup 2013 - Eindhoven



RoboCup 2013
<https://www.flickr.com/photos/robocup2013/10151792836>

<https://creativecommons.org/licenses/by-nc-sa/2.0/>

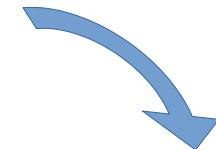
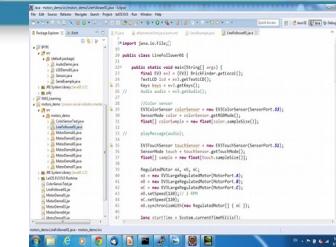
Стилове на учене с роботи



Асимилатор
индукция, теории

Рефлексивно
наблюдение

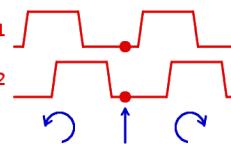
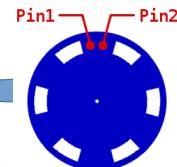
Абстрактна
концептуализация



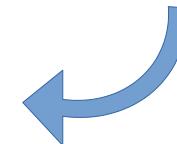
Конвърджър
от общото към частното
прилагане на теориите

Активно
експериментиране

Дивърджър
от частното към общото,
конкретни наблюдения



Конкретен опит



Обучение чрез проекти

Проектно-базираното обучение (PBL) е форма на активно обучение, при която обучаваните получават знания и умения чрез работа и изследване за продължителен период от време върху автентичен, ангажиращ, и сравнително сложен въпрос, проблем или предизвикателство.

-- What is Project Based Learning (PBL)?
https://www.bie.org/about/what_pbl



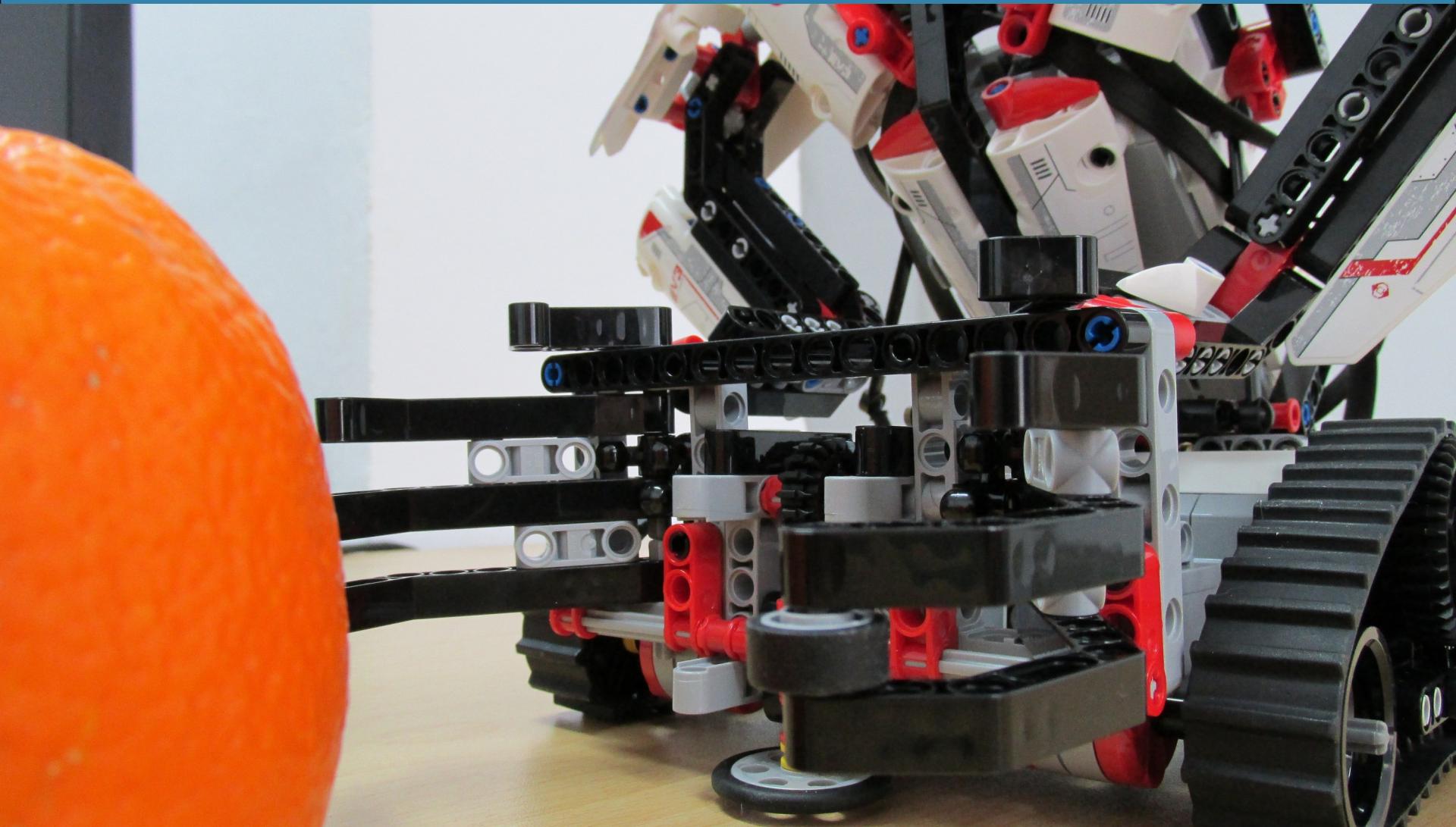
Защо PBL?

[https://www.bie.org/about/what_pbl]

- ❖ Ключови знания, разбиране, критично мислене/решаване на проблеми, комуникация, колаборация, себе-управление и планиране на ресурсите.
- ❖ Предизвикателство с реална ценност – необходимост от адаптиране на сложността на решаваните проблеми.
- ❖ Непрекъснато изследване – продължителен процес на търсене на отговори на въпроси и информационни ресурси, прилагане на информацията на практика
- ❖ Автентичност и релевантност – контекст, задачи, стандарти за качество от реалния свят в съответствие с персоналните интереси на обучаваните
- ❖ Личен избор и креативност – всеки екип сам избира какво ще създаде.
- ❖ Рефлексия – обучаваните и преподавателя обсъждат наученото/ процеса, заедно решават проблемите: решенията не са предварително известни!
- ❖ Обратна връзка – обучаваните дават и получават обратна връзка и непрекъснато усъвършенстват както продукта, така и процеса на работа.
- ❖ Публичен продукт – резултатът се демонстрира публично – пред колеги и в по-широва общност извън класната стая.

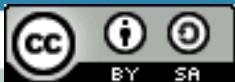


Запознайте се с LeJaRo

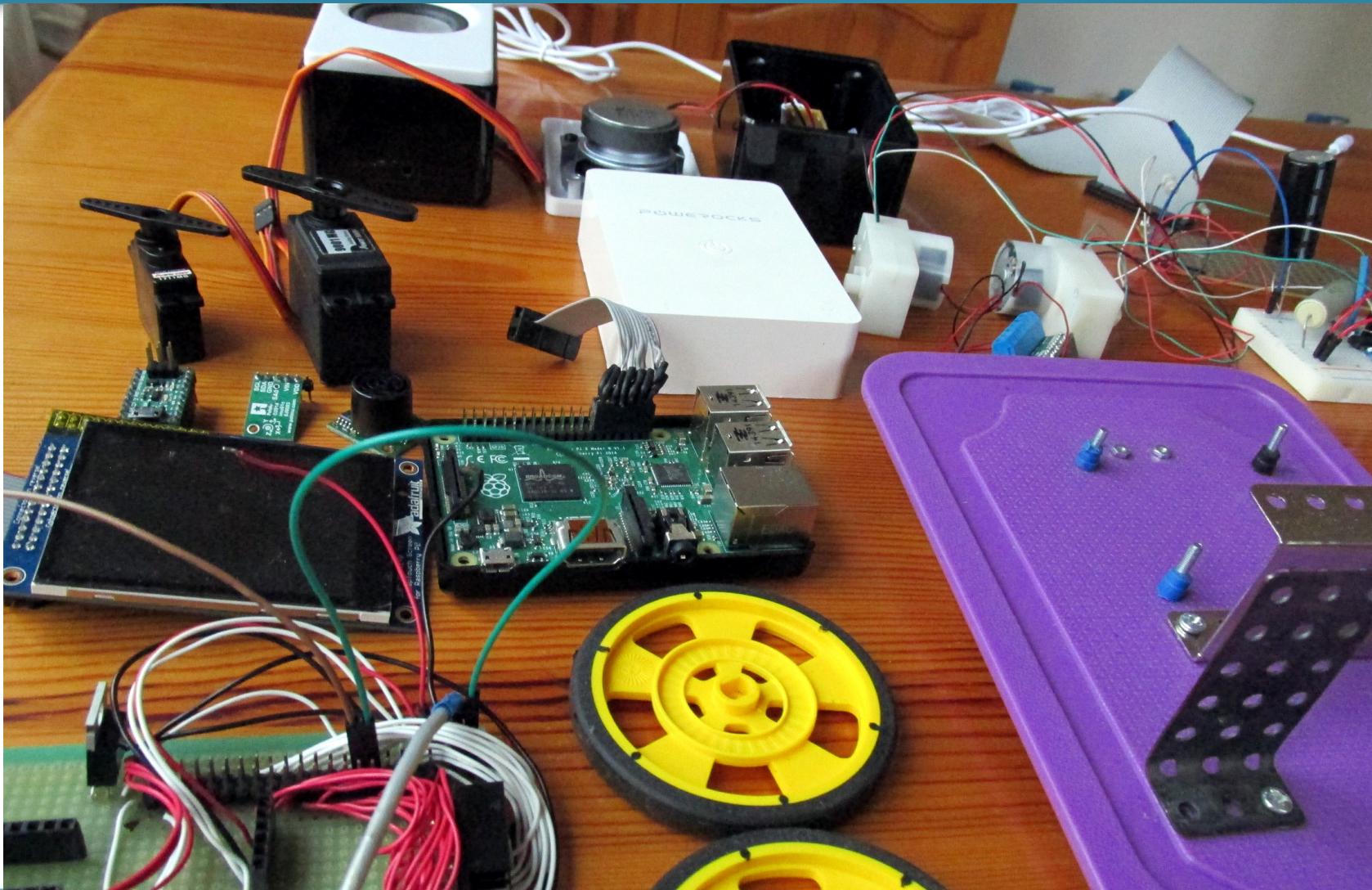


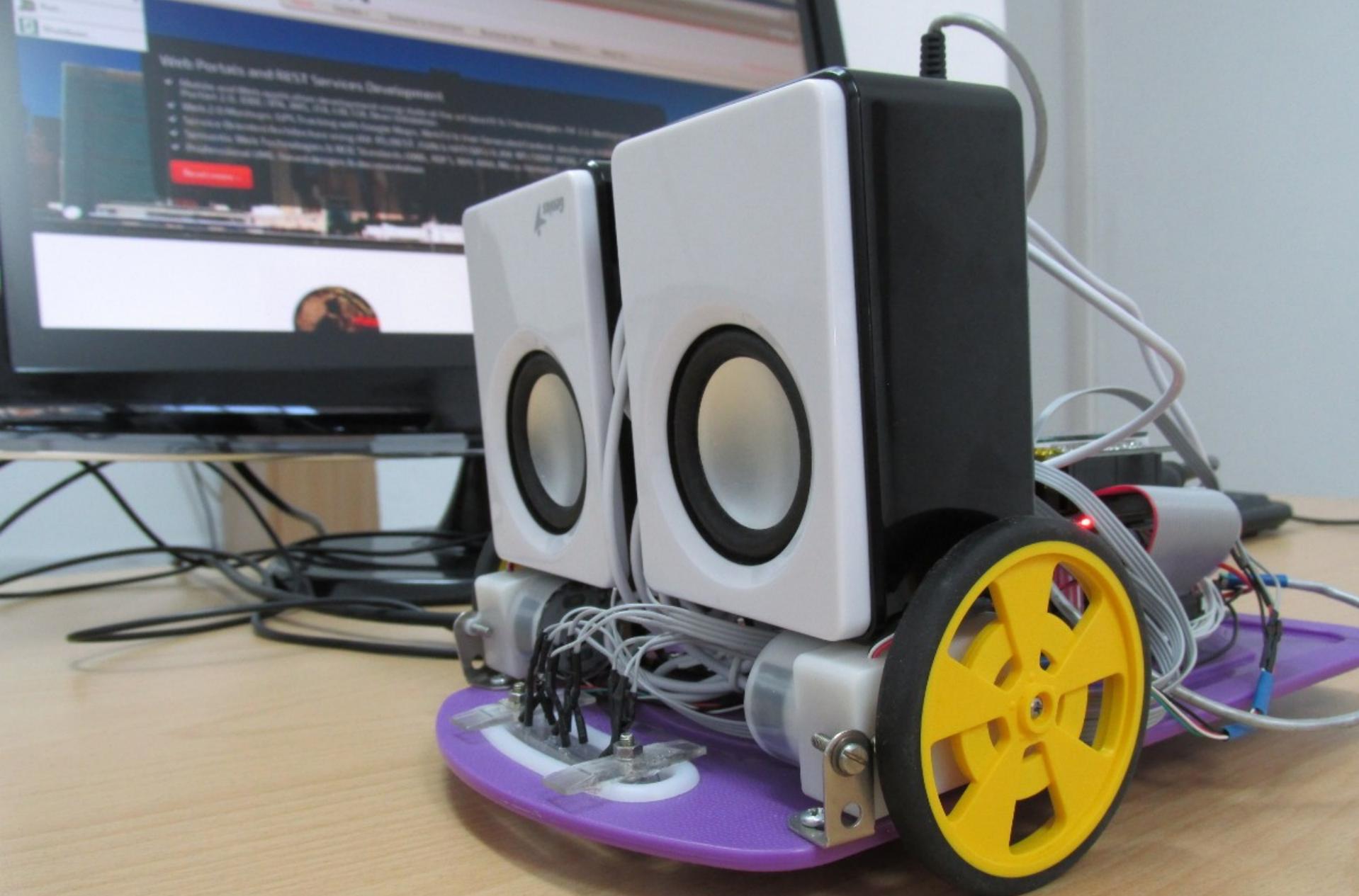
LeJaRo: Lego® Java Robot

- ❖ Modular – 3 *motors (with encoders)* – one driving each track, and third for robot clamp.
- ❖ Three sensors: *touch sensor* (obstacle avoidance), *light color sensor* (follow line), *IR sensor* (remote).
- ❖ LeJaRo is programmed in Java using **LeJOS** library.
- ❖ More information about LeJaRo:
<http://robolearn.org/lejaro/>
- ❖ Programming examples available @GitHub:
https://github.com/iproduct/course-social-robotics/tree/master/motors_demo

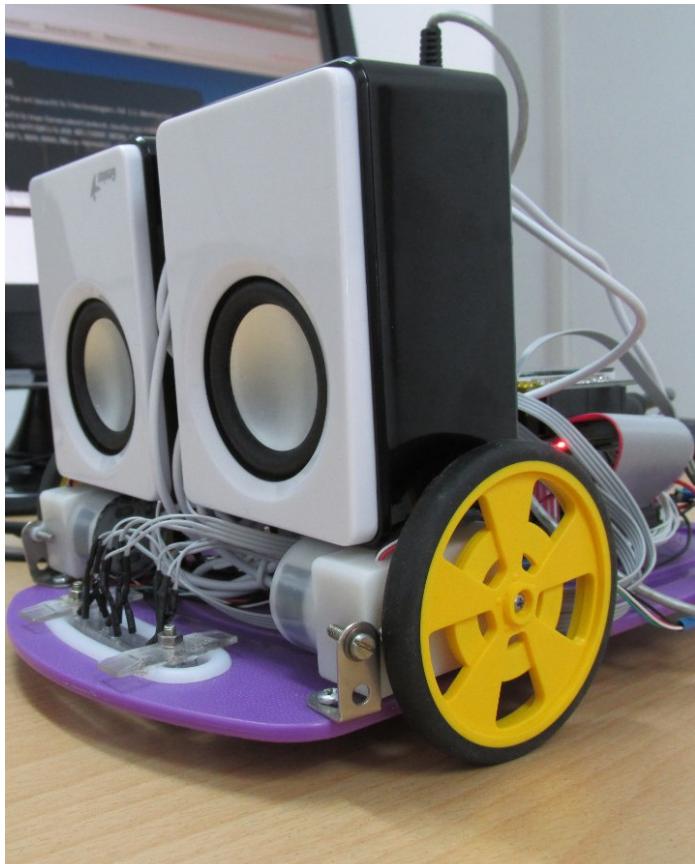


Meet IPTPI :)





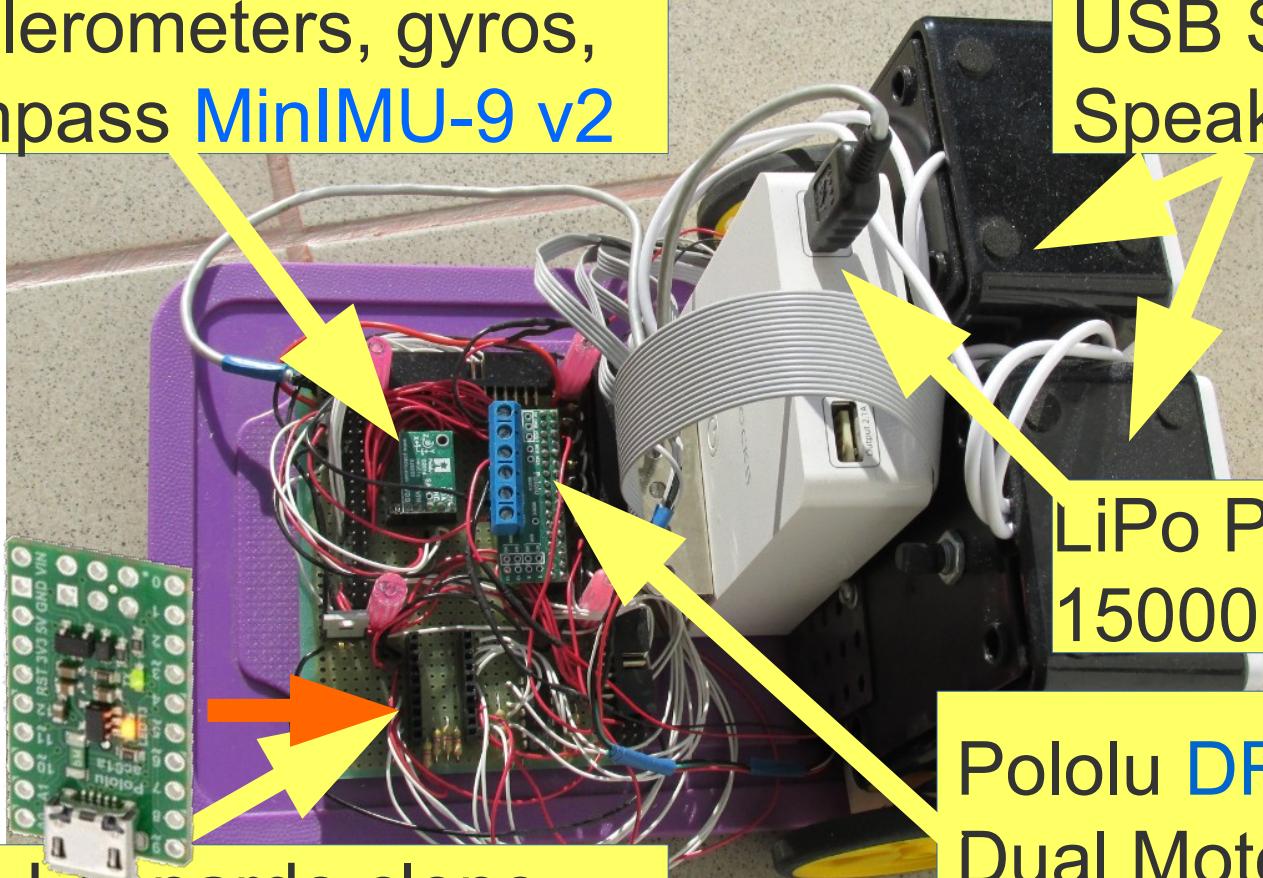
IPTPI: RPi2 + Arduunio Robot



- ❖ Raspberry Pi 2 (quad-core ARMv7 @ 900MHz) + Arduino Leonardo clone **A-Star 32U4 Micro**
- ❖ *Optical encoders* (custom), IR optical array, 3D accelerometers, gyros, and compass **MinIMU-9 v2**
- ❖ **IPTPI** is programmed in Java using **Pi4J**, **Reactor**, **RxJava**, **Akka**
- ❖ More information about IPTPI:
<http://robolearn.org/iptpi-robot/>

IPTPI: RPi2 + Arduinio Robot

3D accelerometers, gyros,
and compass [MinIMU-9 v2](#)



Arduino Leonardo clone
[A-Star 32U4 Micro](#)

USB Stereo
Speakers - 5V

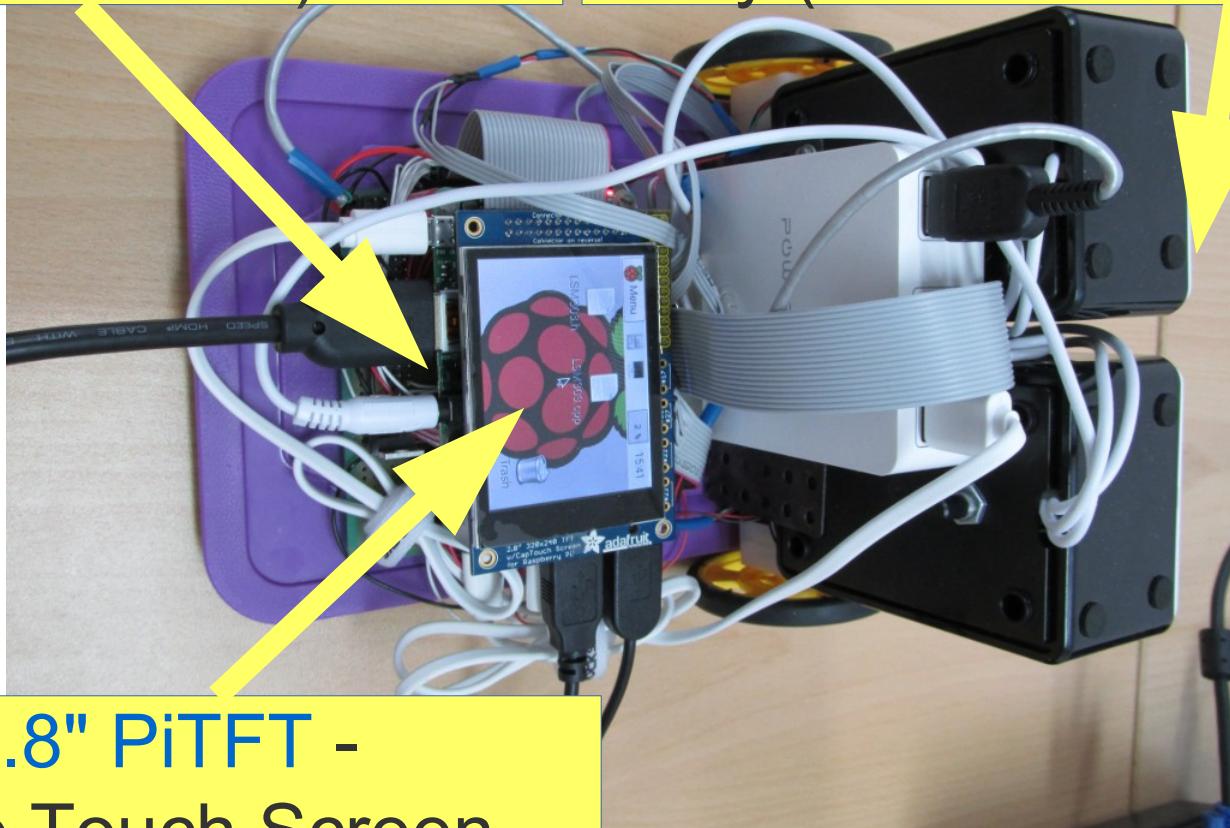
LiPo Powebank
15000 mAh

Pololu [DRV8835](#)
Dual Motor Driver
for Raspberry Pi

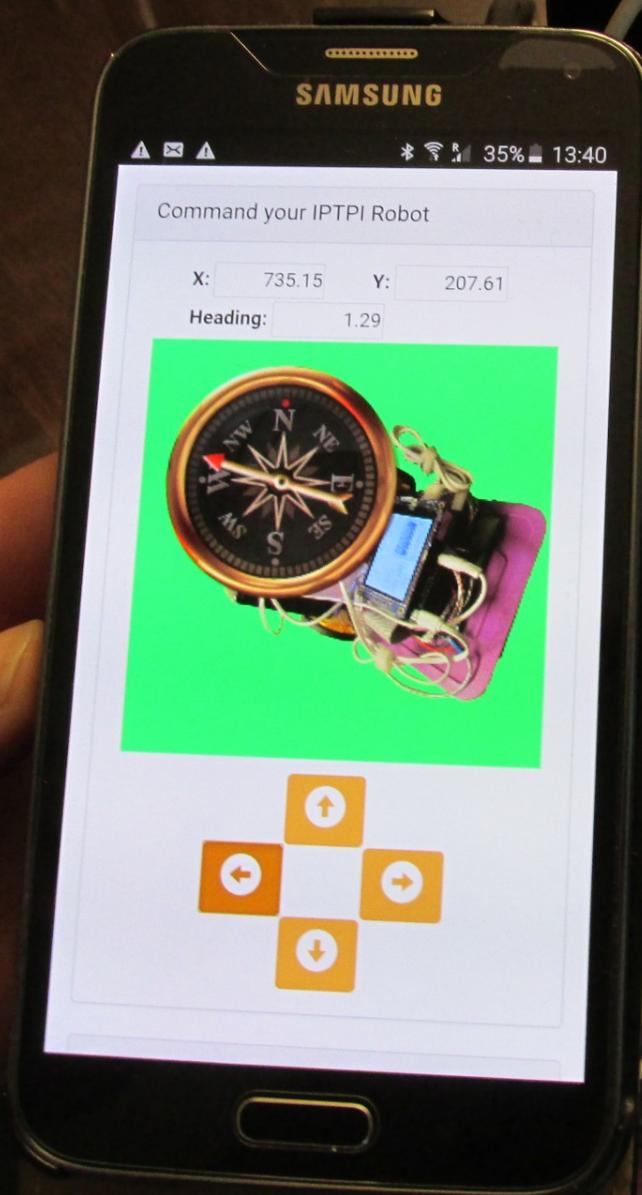
IPTPI: RPi2 + Arduunio Robot

Raspberry Pi 2 (quad-core
ARMv7 @ 900MHz)

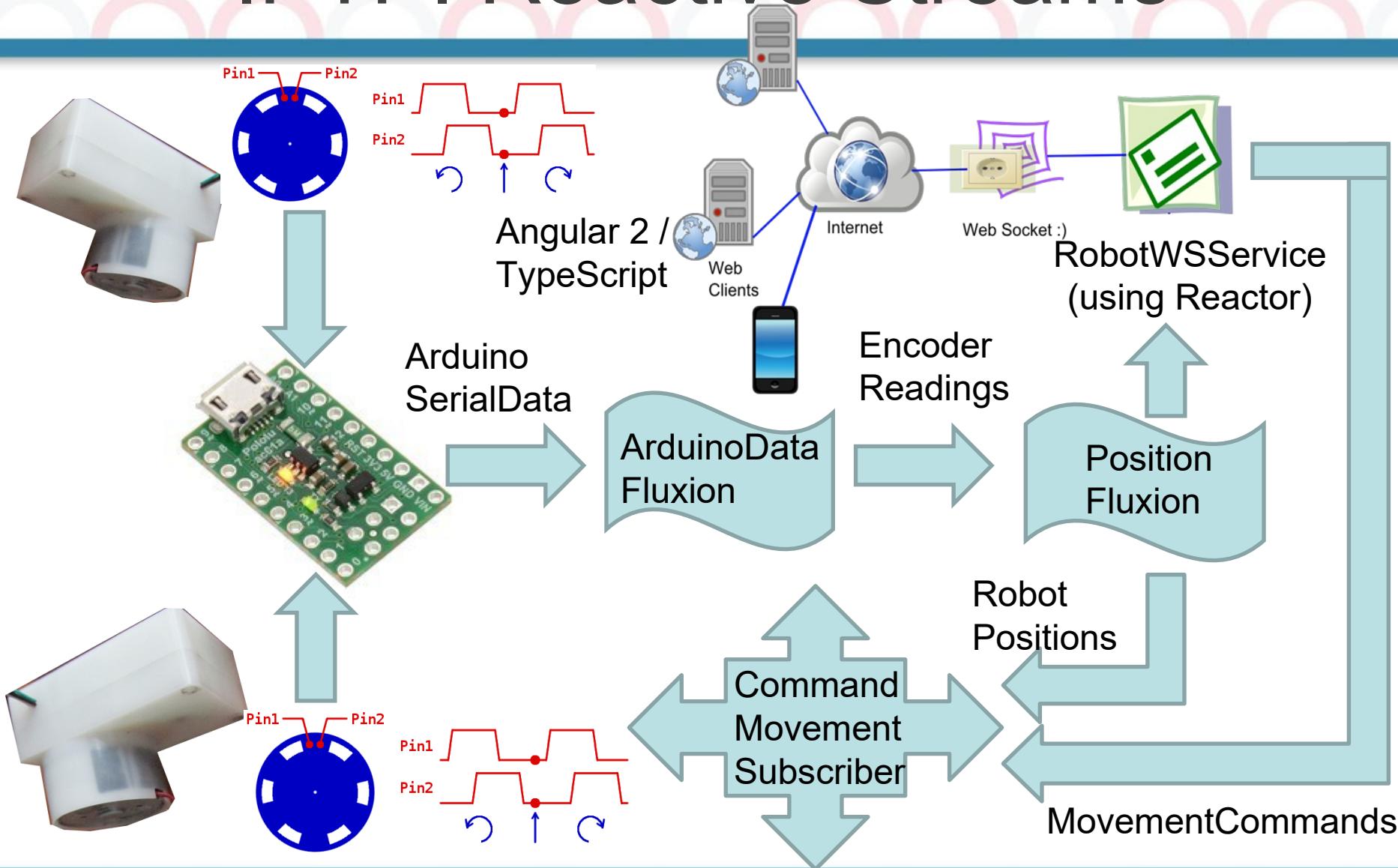
IR Optical Sensor QRD1114
Array (Line Following)

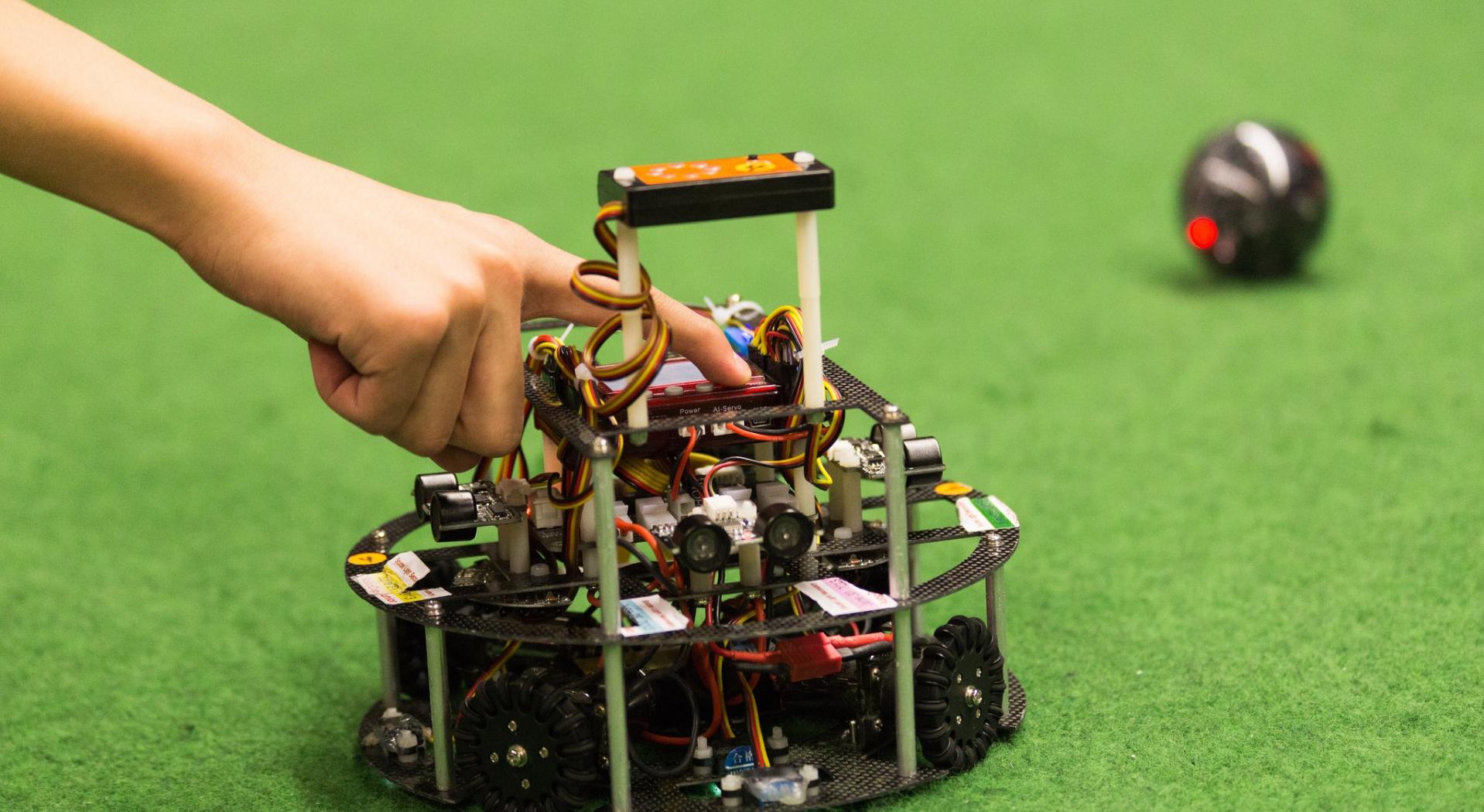


Adafruit 2.8" PiTFT -
Capacitive Touch Screen



IPTPI Reactive Streams





RoboCup 2013 – Eindhoven by Albert van Breemen

[<https://www.flickr.com/photos/robocup2013/10151792836>]

License: Attribution-NonCommercial-ShareAlike 2.0 Generic (CC BY-NC-SA 2.0)

<https://creativecommons.org/licenses/by-nc-sa/2.0/>

I2C, SPI, Serial комуникационни протоколи

- General-purpose input/output (GPIO) – allow digital (and Analog on Arduino) input and output
- Allows listeners to be registered for events connected with the signal
- Supports I²C (Inter-Integrated Circuit) – slower but uses only two pins for bidirectional data transmission with many devices (different addresses) - <https://en.wikipedia.org/wiki/I%C2%B2C>
- Serial Peripheral Interface (SPI) – much faster but requires additional wires - https://en.wikipedia.org/wiki/Serial_Peripheral_Interface
- Pulse-Width Modulation (PWM) –
https://en.wikipedia.org/wiki/Pulse-width_modulation
- See for more details IPT repository @ GitHub:
<https://github.com/iproduct/course-social-robotics/wiki/Study-Materials>

PROGRAMMING PROJECTS



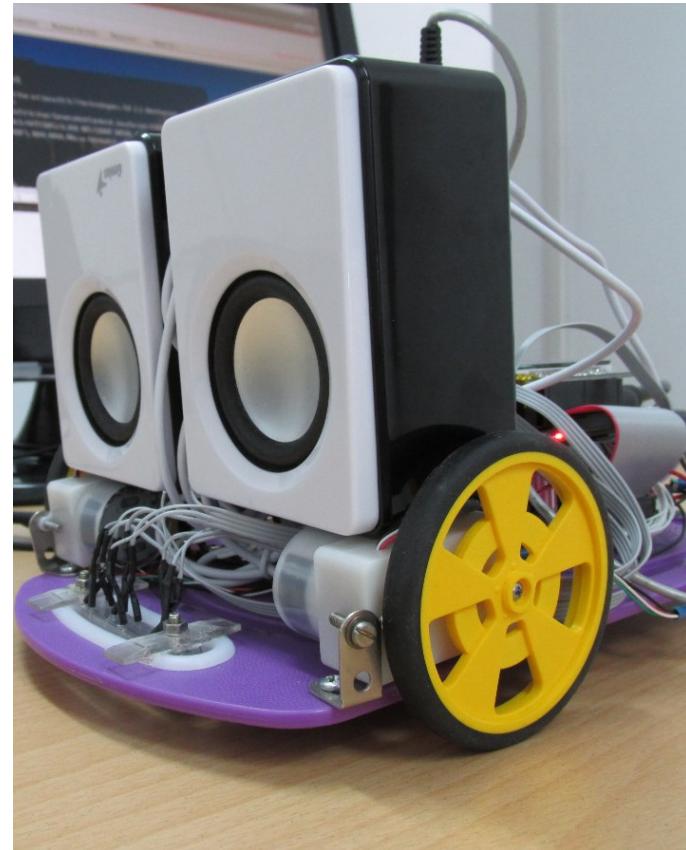
LeJaRo: Follow The Line & Grasp That CUP



- Using **LeJOS APIs** program the robot to follow a **white/color line**.
- When reaching the end of the line LeJaRo should **play a pre-programmed sound**, go **8 cm further**, and **grasp the coffee cup** placed there.
- Then the robot should **turn back** and **go 40 cm strait**, **drop the coffee cup**, and **play another sound** signaling the successful finishing of the task.

IPTPI : Universal Soldier

- Program the robot to say welcome ([GPIO, mplayer + ProcessBuilder](#))
- Read motor encoders data from [Arduino A-Star Micro](#) using [USB serial](#) and [program a feedback loop](#) to straighten the course.
- Read data from [optical array](#) to program robot to [follow a line](#)
- Read compass, accelerometer and gyroscope data ([Pulolu MinIMU-9](#)) and make the robot orient better



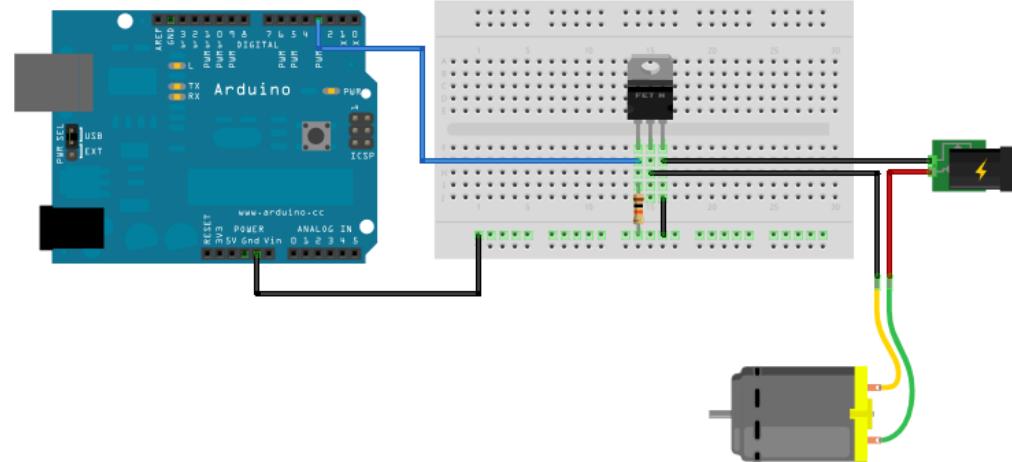
Make Two Wheels Rpi2 Robot from Scratch



Cocktail Maker: Make Your Wish



- Assemble RPi2 / Arduino, low voltage pumps (3x), liquid flow-meters and hoses to a machine able to dose precisely 3 types of liquids
- Assemble breadboard with 3 MOSFET transistors (IRL 540N)



The Platforms

- Lego® Mindstorms® - <http://www.lego.com/en-us/mindstorms>
- Raspberry Pi™ -
<https://www.raspberrypi.org/raspberry-pi-2-on-sale/>
- Banana Pi® - <http://www.bananapi.org/p/product.html>
- SparkFun® pcDuino v2 & v3 -
<https://www.sparkfun.com/products/12856>
- BeagleBone - <http://beagleboard.org/bone>
- ODROID – <http://www.hardkernel.com/main/main.php>
- Arduino (Sorry no Java™ here – YET) & all the clones
- PINGUINO – <http://www.pinguino.org/controllers>

LEGO® is a registered trademark of LEGO® Group. Programs of IPT are not affiliated, sponsored or endorsed by LEGO® Education or LEGO® Group. Raspberry Pi™ is a trademark of Raspberry Pi Foundation. Other names may be trademarks of their respective owners.



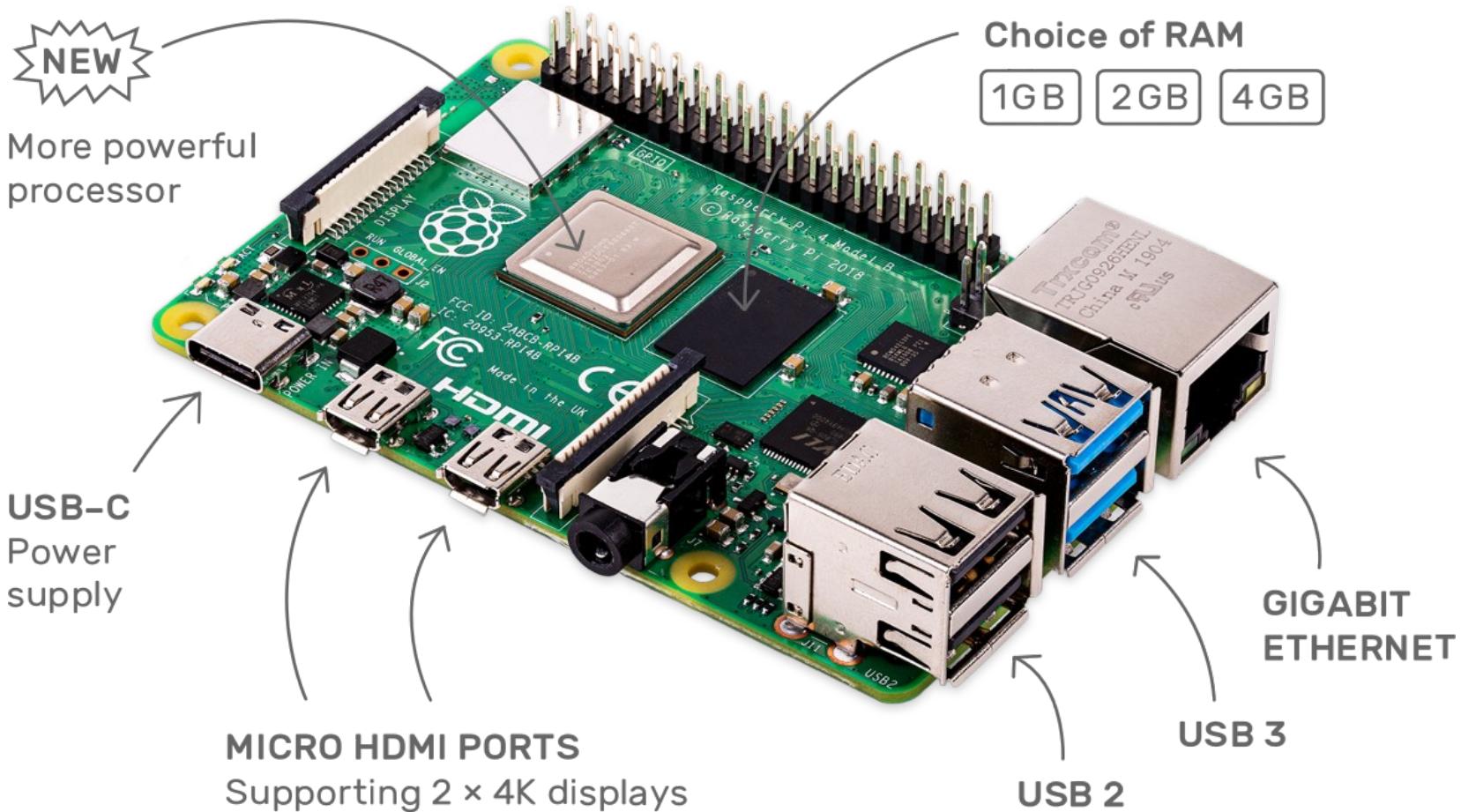
Lego® Mindstorms®

- Pros:
 - Easy to use – even for children, using graphical editor
 - Ready made hardware components – no need to solder and drill :)
 - There is a Java on Lego project – LeJOS (www.lejos.org/)
- Cons:
 - Too easy to use – the graphical editor is not my personal favourite when programming more complex algorithms, or if integration is needed with other libraries (easy with LeJOS)
 - Custom sensors or Arduino integration may be harder to do
 - Limited computational capabilities

LEGO® is a registered trademark of LEGO® Group. Programs of IPT are not affiliated, sponsored or endorsed by LEGO® Education or LEGO® Group. Raspberry Pi™ is a trademark of Raspberry Pi Foundation. Other names may be trademarks of their respective owners.



Raspberry Pi™



BeagleBoard.org > bone

BeagleBone

What is BeagleBone?

Be sure to check out the new **BeagleBone Black**

BeagleBone is an \$89 [MSRF](#), credit-card-sized Linux computer that connects to the Internet and runs software such as Android 4.0 and Ubuntu. With plenty of I/O and processing power for real-time analysis provided by an AM335x 720MHz ARM® processor, BeagleBone can be complemented with cape plug-in boards to augment functionality.



Processor: **AM335x 720MHz ARM Cortex-A8**

- 256MB DDR2 RAM
- 3D graphics accelerator
- ARM Cortex-M3 for power management
- 2x PRU 32-bit RISC CPUs

Connectivity

- USB client: power, debug and device
- USB host
- Ethernet
- 2x 46 pin headers

Software Compatibility

- 4GB microSD card w/ Angstrom Distribution
- Cloud9 IDE on Node.JS w/ BoneScript library

Purchase

Select a distributor to buy ▾

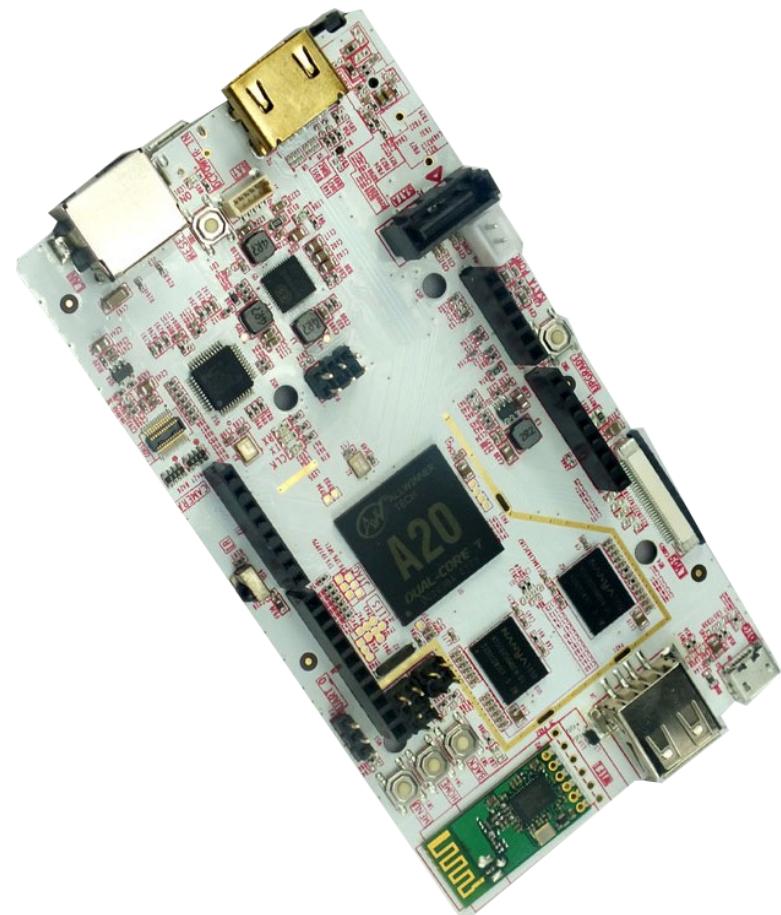
BeagleBoard Projects

BeagleBoard Support

Hardware Specs and Materials

SparkFun® pcDuino v3

- \$59.95, CPU: AllWinner A20 SoC 1GHz ARM Cortex A7 Dual Core
- GPU: OpenGL ES2.0, OpenVG 1.1 Mali 400 Dual Core
- 1GB DRAM
- Onboard Storage: 4GB Flash, microSD card (TF) slot for up to 32GB
- Arduino-Style Peripheral Headers
- SATA Host Socket



ODROID-U3



The Powerful Linux Computer

1.7GHz Quad-Core processor and 2GByte RAM

10/100Mbps Ethernet with RJ-45 LAN Jack

3 x High speed USB2.0 Host ports

Audio codec with headphone jack on board

XUbuntu 13.10 or Android 4.x Operating System

Size : 83 x 48 mm, Weight : 48g including heat sink

Order it now

Price : US\$69.00

[See more detail >>](#)

ODROID-XU3



The Heterogeneous Multi-Processing (HMP) Octa Core Linux Computer

Samsung Exynos5422 Cortex-A15 2.0Ghz quad core and Cortex-A7 quad core

Mali-T628 MP6(OpenGL ES 3.0/2.0/1.1 and OpenCL 1.1 Full profile)

eMMC5.0 HS400 Flash Storage

USB 3.0 Host x 1, USB 3.0 OTG x 1, USB 2.0 Host x 4

XUbuntu 14.04 or Android 4.4 Operating System

Size : 90 x 70 x 18 mm

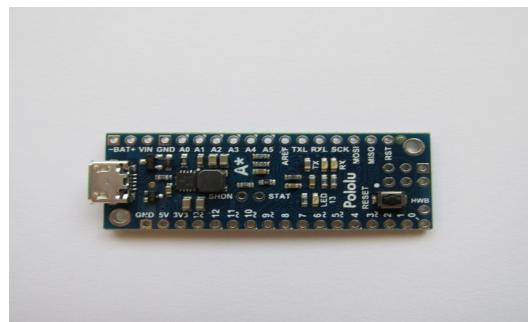
Order it now

Price : US\$179.00

[See more detail >>](#)

Arduino and the Clones

- Open hardware and software project
- Low level and real time microcontroller programming
- Easy to program using Arduino IDE
- New: Plugin for Eclipse under development!
- There is a big variety of boards
- Even very cheap ones < 15\$
- Pulolu
- A-Star
- Micro/Mini



ks Tools Help

BeagleBoard.o... Robotev.com: ... Robotev.com: ... Arduino - ... PIC32-PINGUI... Raspberry Pi 2 ... Banana Pi - A ... ODROID | Hardker... LeJOS, Java f...

ain/Products

arduino ide

LOG IN SIGN UP

Buy Download Products Learning Forum Support Blog

Arduino Uno

Arduino Leonardo

Arduino GSM Shield

The Arduino Starter Kit

Arduino Due

Arduino Yún

Arduino Ethernet Shield

Arduino Materia 101

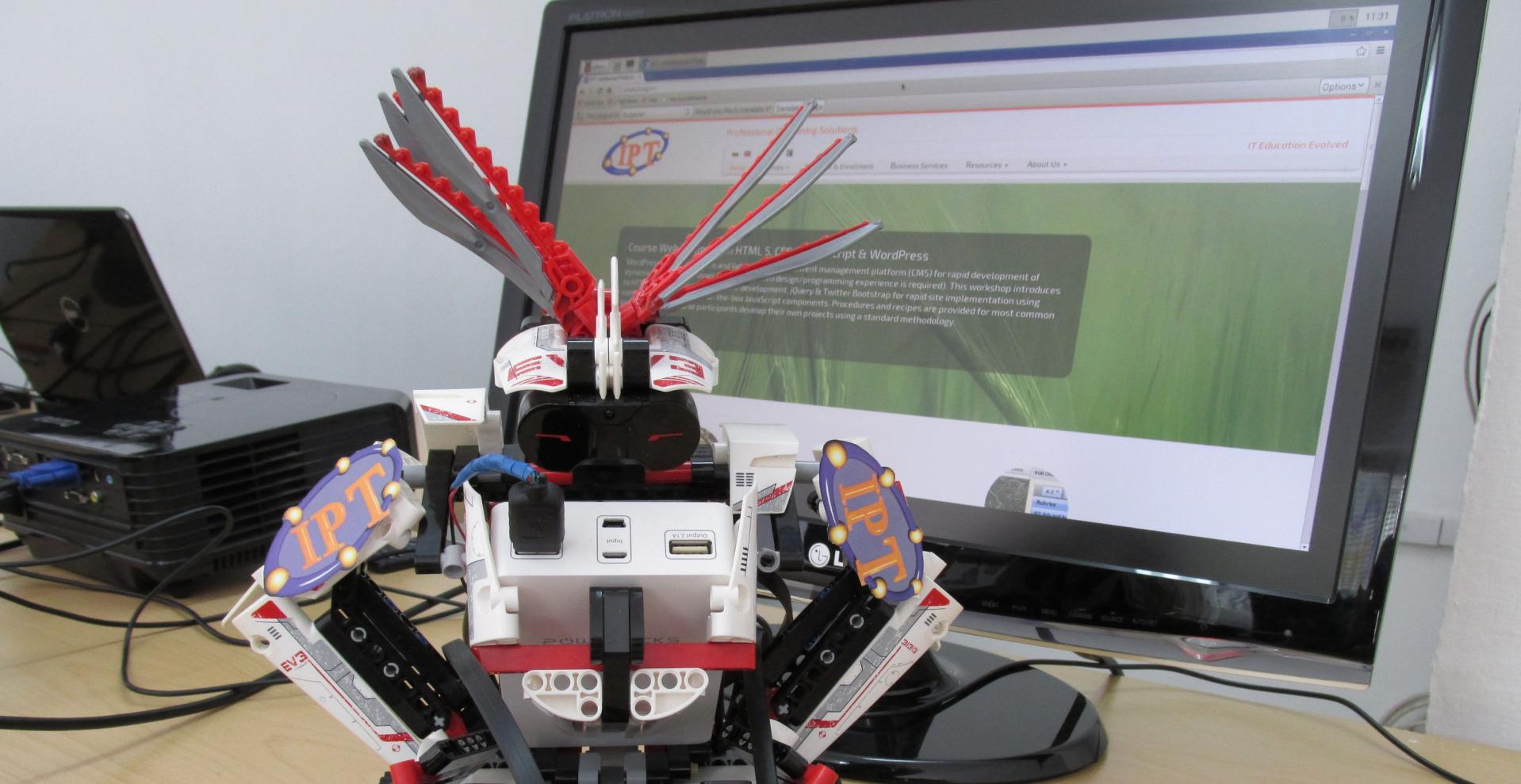
Arduino Tre

Arduino Zero

Arduino WiFi Shield

ACCESSORIES

This screenshot shows the Arduino website's product page. At the top, there's a navigation bar with links for Buy, Download, Products, Learning, Forum, Support, and Blog. On the right, there are LOG IN and SIGN UP buttons. Below the navigation, there are two rows of products. The first row contains four items: Arduino Uno, Arduino Leonardo, Arduino GSM Shield, and The Arduino Starter Kit. The second row contains four items: Arduino Due, Arduino Yún, Arduino Ethernet Shield, and Arduino Materia 101. The third row contains two items: Arduino Tre and Arduino Zero. Both of these have yellow circles with the text 'COMING SOON!' overlaid on them. To the right of the Arduino Zero image, the word 'ACCESSORIES' is written in large, bold, teal letters. At the very bottom, there's a horizontal bar with icons for Google Chrome, File Explorer, Mozilla Firefox, and Internet Explorer, along with an EN link.



But Lets See Some Java Code: LeJOS

```
public class MotorDemo07 {  
    public static void main(String[] args) {  
        final EV3 ev3 = (EV3) BrickFinder.getLocal();  
        TextLCD lcd = ev3.getTextLCD();  
        Audio audio = ev3.getAudio();  
        //Color sensor  
        EV3ColorSensor colorSensor = new  
        EV3ColorSensor(SensorPort.S3);  
        SensorMode color = colorSensor.getRGBMode();  
        float[] colorSample = new  
        float[color.sampleSize()];  
        playMessage(audio);
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** On the left, under the "motors_demo" project, the "src" folder contains several Java files: AudioDemo.java, LEDDemo01.java, Sensors.java, SerialExample.java, LineFollower01.java (selected), MotorDemo01.java, MotorDemo02.java, MotorDemo03.java, MotorDemo04.java, MotorDemo05.java, MotorDemo06.java, MotorDemo07.java, and MotorDemo08.java.
- Code Editor:** The main window displays the content of `LineFollower01.java`. The code initializes EV3 components, sets up sensors for color and touch, and configures regulated motors for a line-following task.
- Toolbars:** Standard Eclipse toolbars for file operations, search, and navigation are visible at the top.
- Quick Access:** A toolbar with icons for Java EE, Java, and Debug.
- Right Sidebar:** Shows the Java EE perspective with various icons for deployment and monitoring.

```
3 import java.io.File;
19
20 public class LineFollower01 {
21
22     public static void main(String[] args) {
23         final EV3 ev3 = (EV3) BrickFinder.getLocal();
24         TextLCD lcd = ev3.getTextLCD();
25         Keys keys = ev3.getKeys();
26         Audio audio = ev3.getAudio();
27
28         //Color sensor
29         EV3ColorSensor colorSensor = new EV3ColorSensor(SensorPort.S3);
30         SensorMode color = colorSensor.getRGBMode();
31         float[] colorSample = new float[color.sampleSize()];
32
33         playMessage(audio);
34
35         EV3TouchSensor touchSensor = new EV3TouchSensor(SensorPort.S1);
36         SensorMode touch = touchSensor.getTouchMode();
37         float[] sample = new float[touch.sampleSize()];
38
39         RegulatedMotor mA, mB, mC;
40         mA = new EV3LargeRegulatedMotor(MotorPort.A);
41         mB = new EV3LargeRegulatedMotor(MotorPort.B);
42         mC = new EV3LargeRegulatedMotor(MotorPort.C);
43         mB.setSpeed(120); // 2 RPM
44         mC.setSpeed(120);
45         mB.synchronizeWith(new RegulatedMotor[] { mC });
46
47         long startTime = System.currentTimeMillis();
```

LeJOS (2)

```
EV3TouchSensor touchSensor = new  
    EV3TouchSensor(SensorPort.S1);  
SensorMode touch = touchSensor.getTouchMode();  
float[] sample = new float[touch.sampleSize()];
```

```
RegulatedMotor mA, mB, mC;  
mA = new EV3LargeRegulatedMotor(MotorPort.A);  
mB = new EV3LargeRegulatedMotor(MotorPort.B);  
mC = new EV3LargeRegulatedMotor(MotorPort.C);  
mB.setSpeed(120); // 2 RPM  
mC.setSpeed(120);
```

LeJOS (3)

```
mB.synchronizeWith(new RegulatedMotor[] { mC });

long startTime = System.currentTimeMillis();

long duration;

int lastColorId = Color.NONE;

for (int i = 0; i < 4; i++) {
    // go forward

    mB.startSynchronization();

    mB.forward();

    mC.forward();

    mB.endSynchronization();
```

LeJOS (4)

```
// go until not obstacle
do {
    duration = System.currentTimeMillis() - startTime;
    touch.fetchSample(sample, 0);
    color.fetchSample(colorSample, 0);
    lcd.drawString(" " + colorSample[0], 0, 3);
    lcd.drawString(" " + colorSample[1], 0, 4);
    lcd.drawString(" " + colorSample[2], 0, 5);
} while (duration < 60000 && mB.isMoving() &&
        mC.isMoving()
        && sample[0] == 0 && isReflecting(colorSample));
```

LeJOS (5)

```
mB.startSynchronization(); // go back  
mB.backward();  
mC.backward();  
mB.endSynchronization();  
mB.startSynchronization(); // turn back  
mB.rotate(1000, true);  
mC.rotate(-1000, true);  
mB.endSynchronization();  
while (mB.isMoving() && mC.isMoving())  
    Thread.yield();
```

LeJOS (6)

```
mB.flt();  
mC.flt();  
mA.close();  
mB.close();  
mC.close();  
}
```

```
private static void playMessage(final Audio audio) {  
    audio.playSample(new File("pozdrav_01.wav"), 100);  
}
```

LeJOS (7)

- You have to adjust the thresholds by trial and error :)

```
private static boolean isReflecting(float[] colorSample){  
    return      colorSample[0] > 0.015  
            || colorSample[1] > 0.015  
            || colorSample[2] > 0.015;  
}
```

Pi4J – Java Low Level GPIO (I2C, SPI, Serial) for Raspberry Pi

- General-purpose input/output (GPIO) – allow digital (and Analog on Arduino) input and output
- Allows listeners to be registered for events connected with the signal
- Supports I²C (Inter-Integrated Circuit) – slower but uses only two pins for bidirectional data transmission with many devices (different addresses).
- Serial Peripheral Interface (SPI) – much faster but requires additional wires
- See for more details IPT repository @ GitHub:

<https://github.com/iproduct/course-social-robotics/wiki/Lectures>

Low Level GPIO Programming is Fun :)

```
public class MotorDemo01 {  
  
    public static void main(String[] args) throws InterruptedException {  
  
        // initialize wiringPi library  
        Gpio.wiringPiSetupGpio();  
        // Motor direction pins  
        Gpio.pinMode(5, Gpio.OUTPUT);  
        Gpio.pinMode(6, Gpio.OUTPUT);  
  
        // SoftPwm.softPwmCreate(12, 0, 100);  
        Gpio.pinMode(12, Gpio.PWM_OUTPUT);  
        Gpio.pinMode(13, Gpio.PWM_OUTPUT);  
        Gpio.pwmSetMode(Gpio.PWM_MODE_MS);  
        Gpio.pwmSetRange(480);  
        Gpio.pwmSetClock(2);  
        Gpio.pwmWrite(12, 460);  
        Gpio.pwmWrite(13, 460);  
        Gpio.digitalWrite(5, 0);  
        Gpio.digitalWrite(6, 1);  
        Thread.sleep(5000);  
  
        System.out.println("Running motors forward accelerating");  
    }  
}
```

Pi4J (1) – GPIO Initialization

```
public class MotorDemo01 {  
  
    public static void main(String[] args) throws  
    InterruptedException {  
  
        // initialize wiringPi library using Broadcom  
        pin numbering scheme  
  
        Gpio.wiringPiSetupGpio();
```

Pi4J (2) - Setup

```
// Motor direction pins  
Gpio.pinMode(5, Gpio.OUTPUT);  
Gpio.pinMode(6, Gpio.OUTPUT);  
// Motor speed pulse width modulated (PWM)  
pins  
Gpio.pinMode(12, Gpio.PWM_OUTPUT);  
Gpio.pinMode(13, Gpio.PWM_OUTPUT);  
Gpio.pwmSetMode(Gpio.PWM_MODE_MS);  
Gpio.pwmSetRange(480);  
Gpio.pwmSetClock(2);
```

Pi4J (3) – Turning Left / Right

```
System.out.println("Turning left");
//setting motor directions
Gpio.digitalWrite(5, 1);
Gpio.digitalWrite(6, 0);
//setting speed
Gpio.pwmWrite(12, 460); // speed 460 of 480
max
Gpio.pwmWrite(13, 460);
// turn duration
Thread.sleep(3000);
```

Pi4J (4) – Accelerating / Decelerating

```
System.out.println("Motors forward  
accelerating");  
for (int i = 0; i <= 480; i++) {  
    //setting motor directions  
    Gpio.digitalWrite(5, 1);  
    Gpio.digitalWrite(6, 1);  
    //setting speed  
    Gpio.pwmWrite(12, i);  
    Gpio.pwmWrite(13, i);  
    Thread.sleep(40);  
}
```

Pi4J (5) – Finishing Politely

```
// turning the motors off  
Gpio.digitalWrite(5, 0);  
Gpio.digitalWrite(6, 0);  
Gpio.pwmWrite(12, 0);  
Gpio.pwmWrite(13, 0);  
  
System.out.println("End of the demo.");  
}  
}
```

Pi4J – Java Low Level GPIO (I2C, SPI, Serial) for Raspberry Pi (2)

- More programming examples – on the Pi :)

References (1)

- Social Robotics Lectures (Github wiki) –
<https://github.com/iproduct/course-social-robotics/wiki/Lectures>
- Iliev, T., Ontology Based Multi-Agent Architecture for Adaptive Courseware Delivery, Int. Workshop on Interactive Computer-aided Learning ICL'2003, Villach, Austria, September 24-26, 2003
- Internet of Things (IoT) in Wikipedia –
http://en.wikipedia.org/wiki/Internet_of_Things
- Brooks, R.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, 2(1), 1986, pp. 14-23
- Brooks, R., Intelligence without representation, Artificial Intelligence, 47, 1991, pp.139-159

References (2)

- Iliev, T., Ontology Based Multi-Agent Architecture for Adaptive Courseware Delivery, Int. Workshop on Interactive Computer-aided Learning ICL'2003, Villach, Austria, September 24-26, 2003
- Iliev, T., E-ACADEMIA: Adaptive Courseware Authoring and Delivery Using Learning Support Agents, Int. Workshop on Interactive Computer-aided Learning ICL'2003, Villach, Austria, September 28-30, 2005.
- Russell, S.; Norvig, P.: Artificial Intelligence: A Modern Approach, Prentice Hall, 1995 – Ch. 2:
<http://www.cs.berkeley.edu/~russell/aima1e/chapter02.pdf>

References (3)

- Jennings, N.; Sycara, K.; Wooldridge, M.: A roadmap of agent research and development, *Autonomous Agents and Multi-Agent Systems*, 1 (1), 1998, pp.7-38
- Weiss, G. (editor): *Multiagent Systems: A Modern Approach to Distributed Intelligence*, MIT Press, 2000
- Brooks, R.: A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 1986, pp. 14-23
- Brooks, R., Intelligence without representation, *Artificial Intelligence*, 47, 1991, pp.139-159

Thank's for Your Attention!



Trayan Iliev

CEO of IPT – Intellectual Products
& Technologies

<http://iproduct.org/>

<https://github.com/iproduct>

<https://twitter.com/trayaniliev>

<https://www.facebook.com/IPT.EACAD>