

Android, Linux and Arduino in a tiny single-board computer

Giovanni Burresi

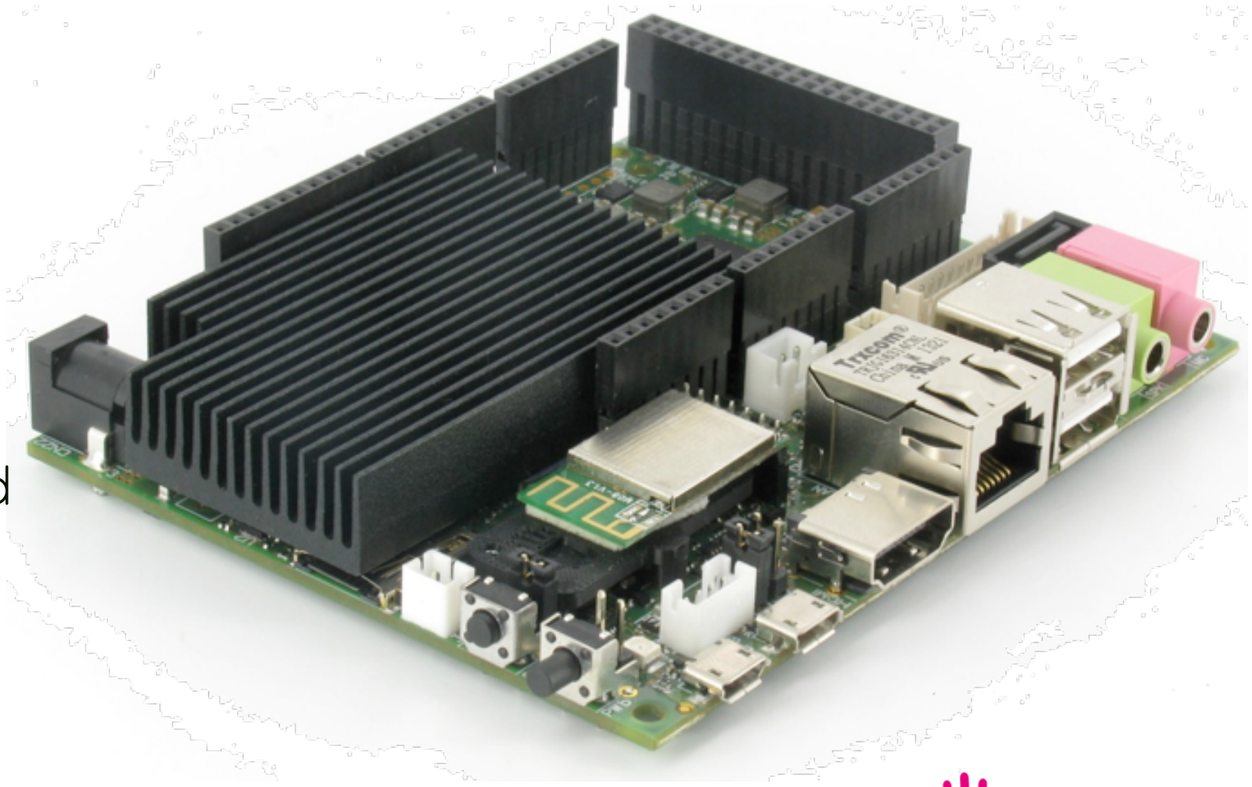
www.udoo.org

What's UDOO?

UDOO is a mini PC that could run either **Android** or **Linux**, with an **Arduino**-compatible board embedded.

UDOO is a collective effort of a multidisciplinary team spread between North America and Europe, with expertise in interaction design, embedded electronics and sensor networks.

UDOO is a co-founded project by SECO (www.seco.com) and Aidilab (www.aidilab.com).

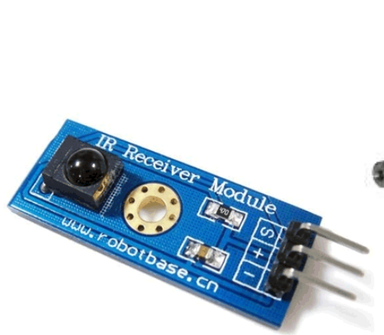


www.udoo.org

Interfacing real-physical world



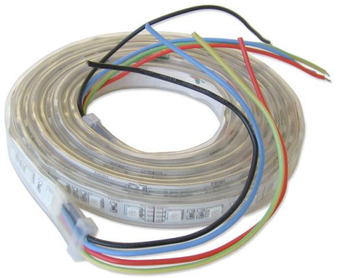










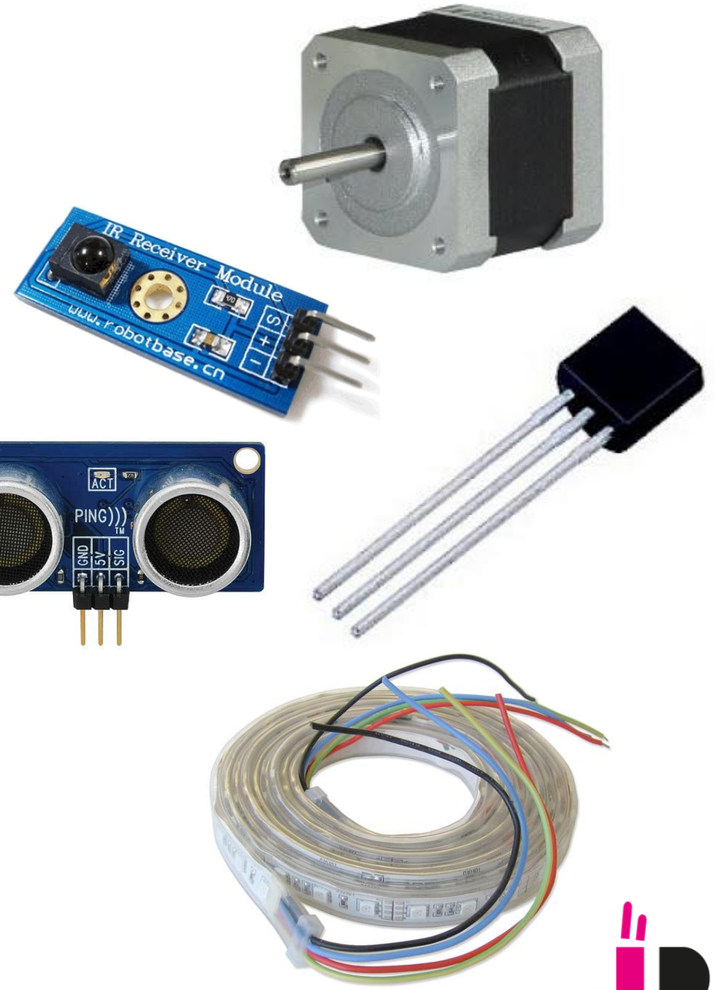


www.udoo.org

Interfacing real-physical world





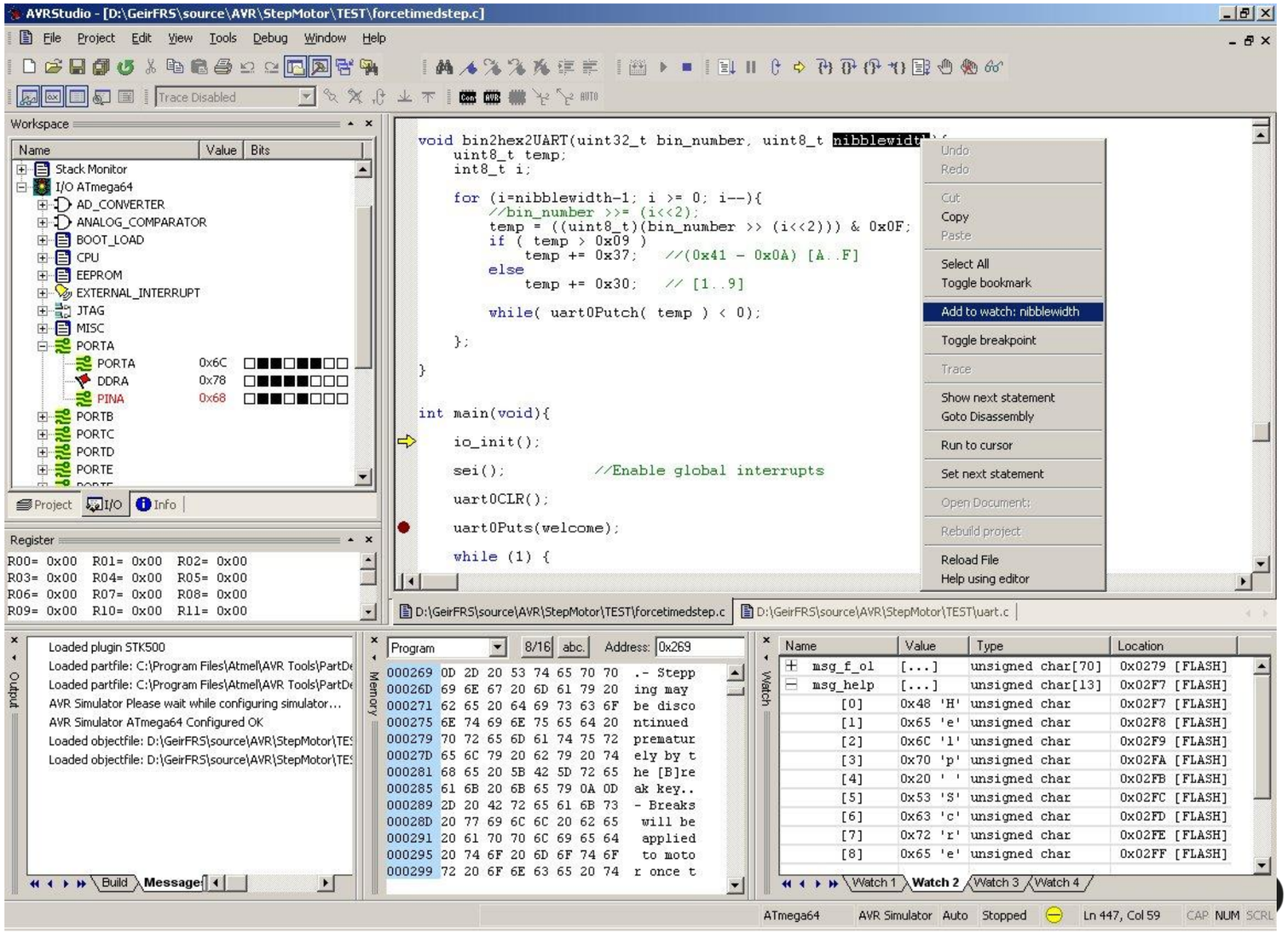


www.udoo.org

Programming microcontrollers

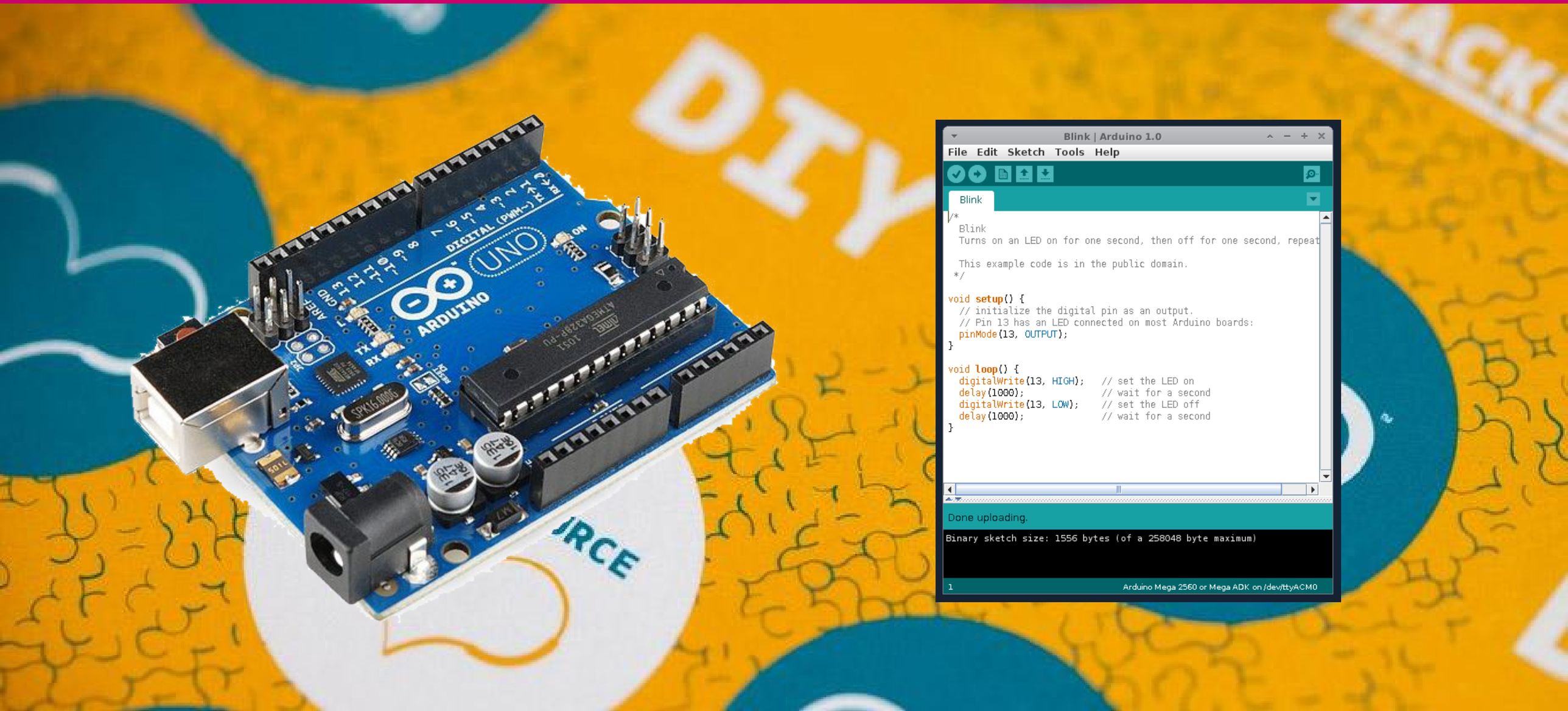
- Complex Dev Environment
- Low level C
- Register programming
- Memory addressing
- Protocols implementation





www.udoo.org

Arduino


Blink | Arduino 1.0
File Edit Sketch Tools Help
Blink
/*
Blink
Turns on an LED on for one second, then off for one second, repeat
This example code is in the public domain.
*/
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected on most Arduino boards:
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH); // set the LED on
delay(1000); // wait for a second
digitalWrite(13, LOW); // set the LED off
delay(1000); // wait for a second
}
Done uploading.
Binary sketch size: 1556 bytes (of a 258048 byte maximum)
1
Arduino Mega 2560 or Mega ADK on /dev/ttyACM0
UNO
ARDUINO
DIGITAL (PWM~)
ON
TX
RX
DIY
RCE

A bit of history

Arduino started in 2005 as a project for students at the Interaction Design Institute Ivrea in Ivrea, Italy. At that time program students used a "BASIC Stamp" at a cost of $100, considered expensive for students. Massimo Banzi, one of the founders, taught at Ivrea.

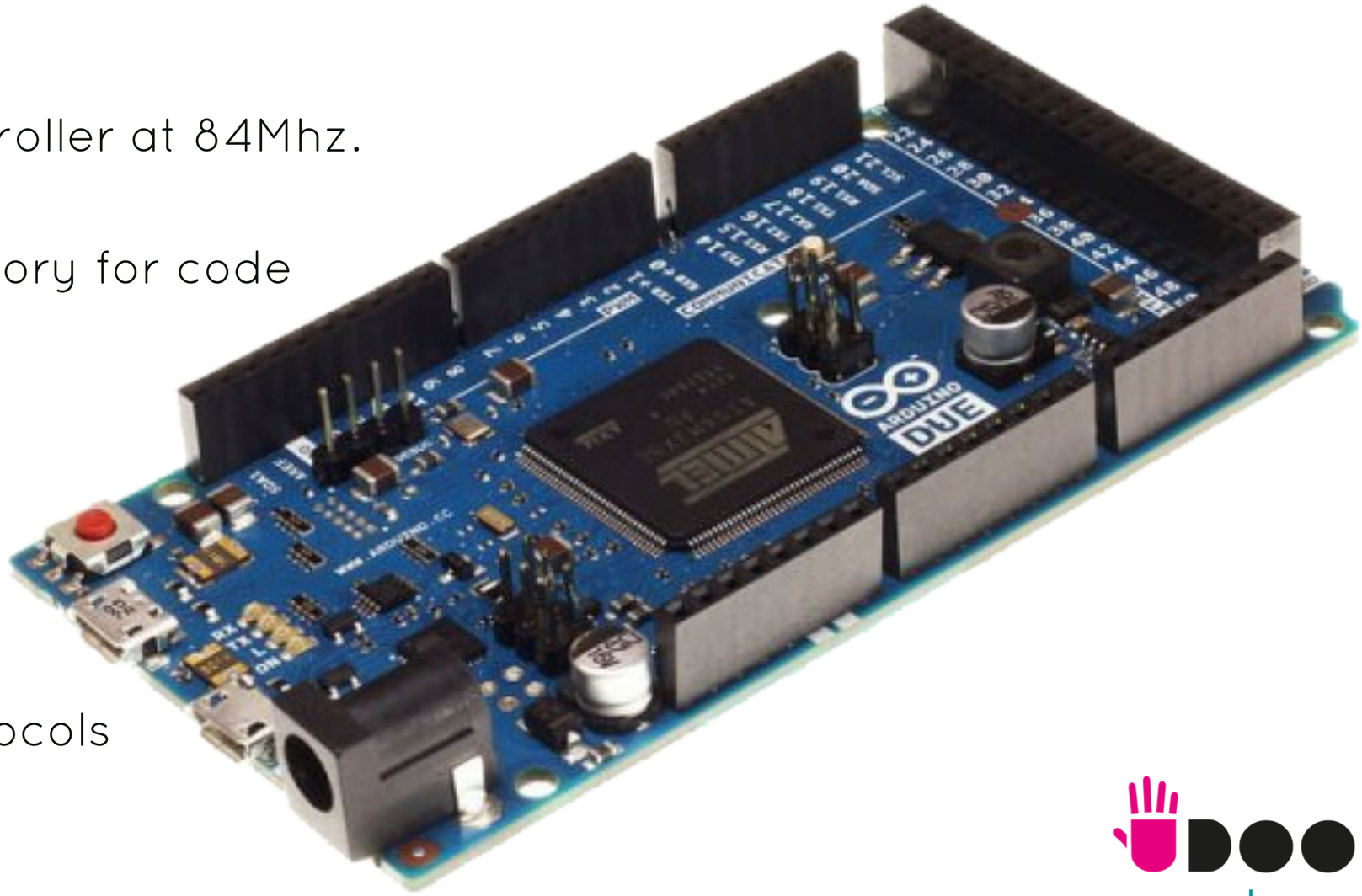
A hardware thesis was contributed for a wiring design by Colombian student Hernando Barragan. After the wiring platform was complete, researchers worked to make it lighter, less expensive, and available to the open source community. The school eventually closed down, so these researchers, one of them David Cuartielles, promoted the idea.

The current prices run around $30 and related "clones" as low as $9.A simple Arduino Mini Pro clone may be had from China for less than $4, post paid.

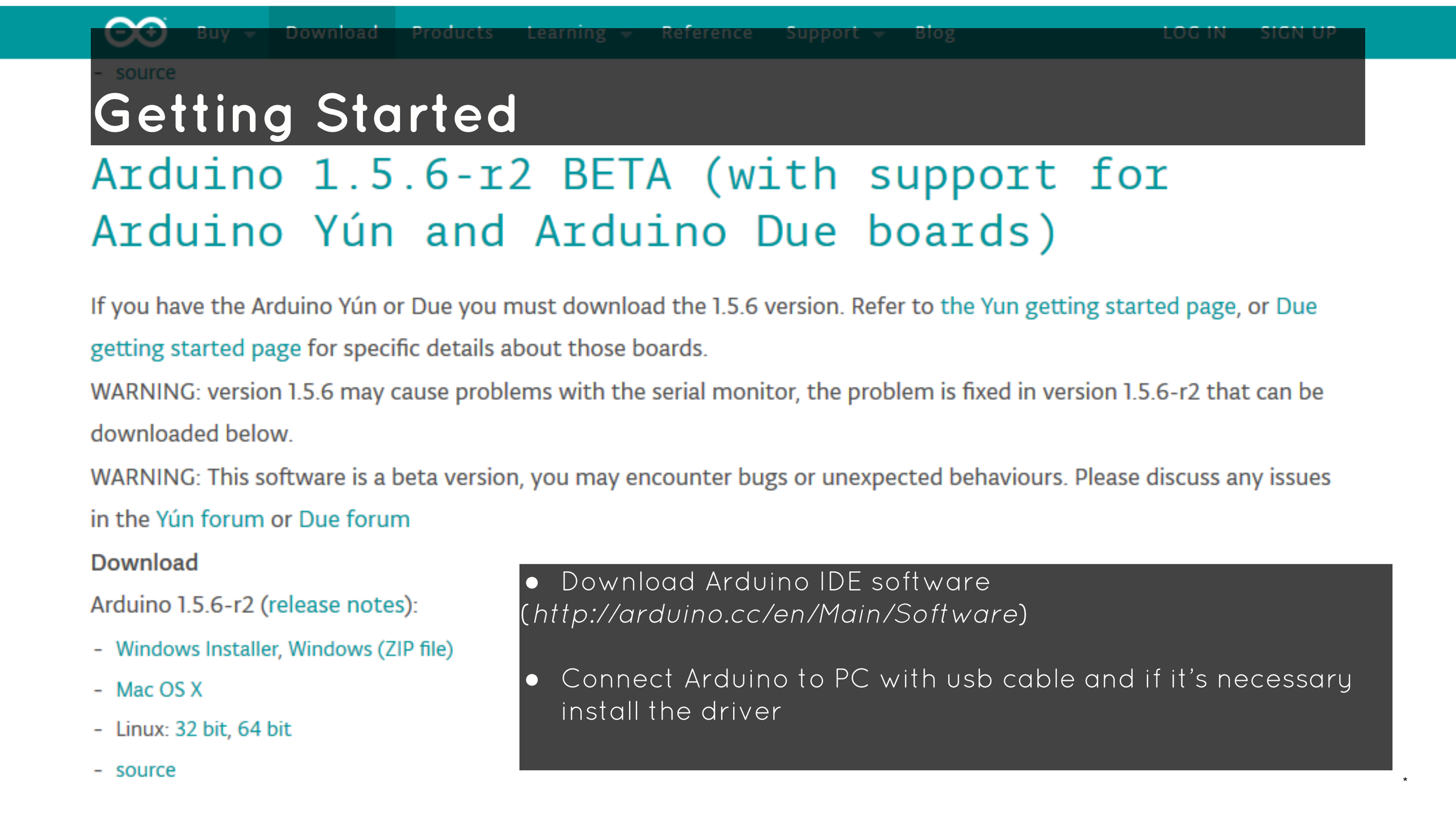
Our choose

- **Atmel SAM3X** microcontroller at 84Mhz.
- 96 KBytes of SRAM.
- 512 KBytes of Flash memory for code
- 3.3 Volts compliant

- 54 Digital I/O (12 PWM)
- 12 analog inputs
- 2 DAC analog outputs
- 4 UARTS, 2 I2C, SPI protocols



www.udoo.org


Buy
Download
Products
Learning
Reference
Support
Blog
LOG IN
SIGN UP
- source
Getting Started
Arduino 1.5.6-r2 BETA (with support for Arduino Yún and Arduino Due boards)
If you have the Arduino Yún or Due you must download the 1.5.6 version. Refer to the Yun getting started page, or Due getting started page for specific details about those boards.
WARNING: version 1.5.6 may cause problems with the serial monitor, the problem is fixed in version 1.5.6-r2 that can be downloaded below.
WARNING: This software is a beta version, you may encounter bugs or unexpected behaviours. Please discuss any issues in the Yún forum or Due forum
Download
Arduino 1.5.6-r2 (release notes):
- Windows Installer, Windows (ZIP file)
- Mac OS X
- Linux: 32 bit, 64 bit
- source
• Download Arduino IDE software (http://arduino.cc/en/Main/Software)
• Connect Arduino to PC with usb cable and if it's necessary install the driver

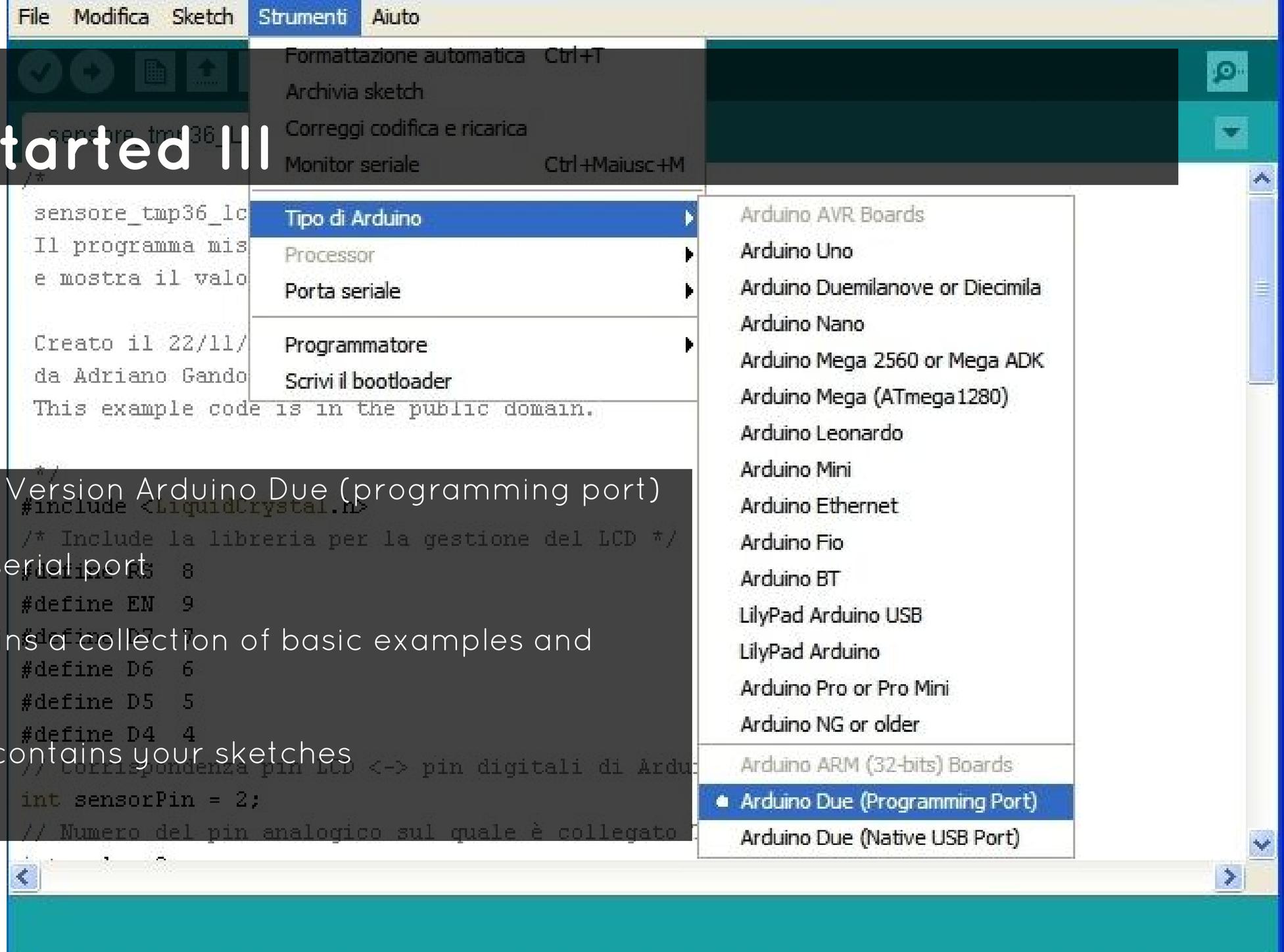
Getting started II - overview



- **compile and verify** the code, from now on we'll call it **sketch**
- **complie** and send to the board the binary
- open a **new** window with empty sketch
- **load** a file from sketch folder
- **save** the sketch
- open a monitor on selected **serial** port

Getting Started III

- Select correct Arduino Version Arduino Due (programming port)
- Select corresponding serial port
- **File -> Examples** contains a collection of basic examples and libraries examples
- **File -> Sketch Folders** contains your sketches



Blink a LED

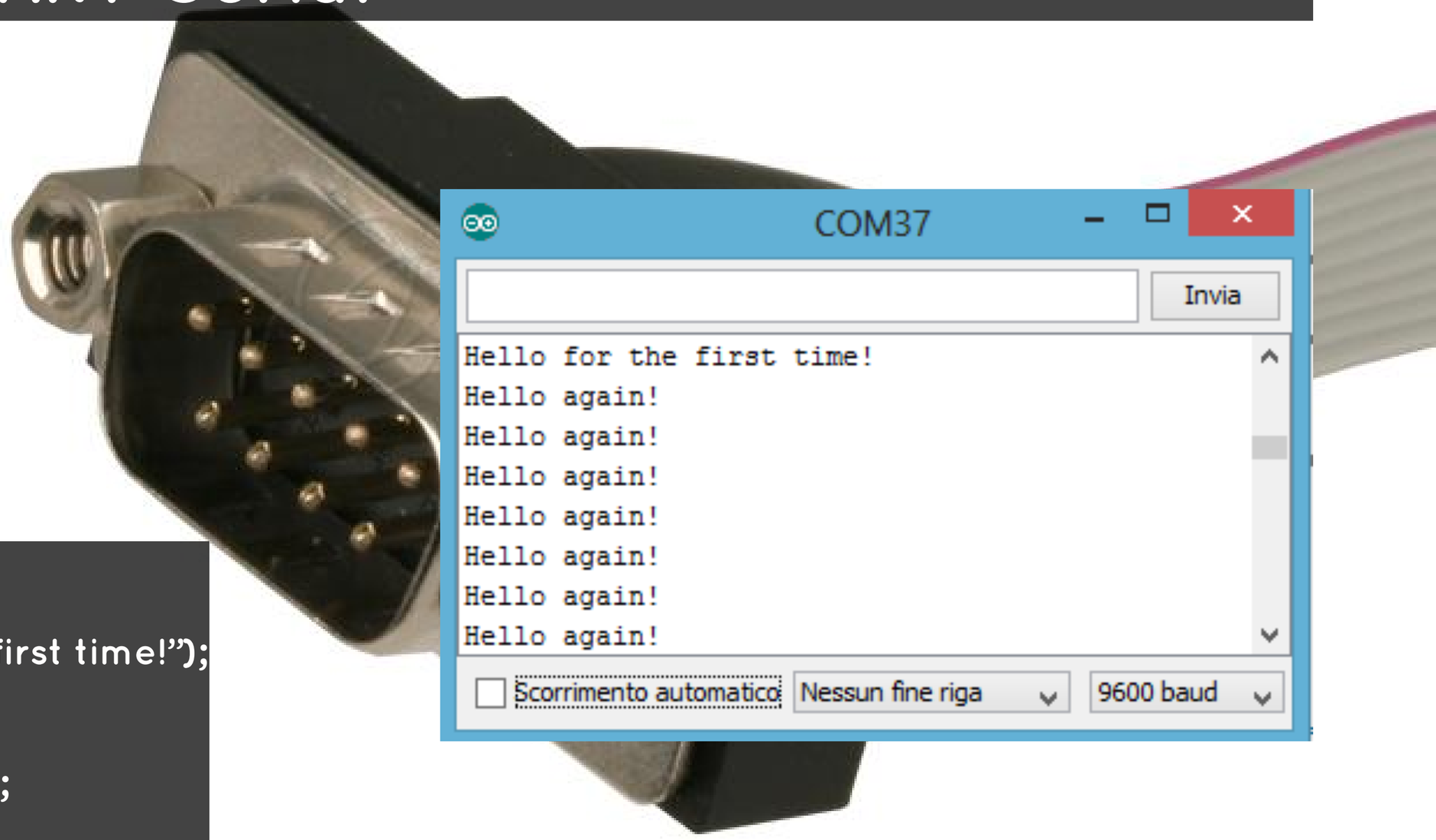


```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Write on UART Serial




```
void setup() {
  Serial.begin(9600);
  Serial.println("Hello for the first time!");
}
void loop() {
  Serial.println("Hello again!");
  delay(1000);
}
```

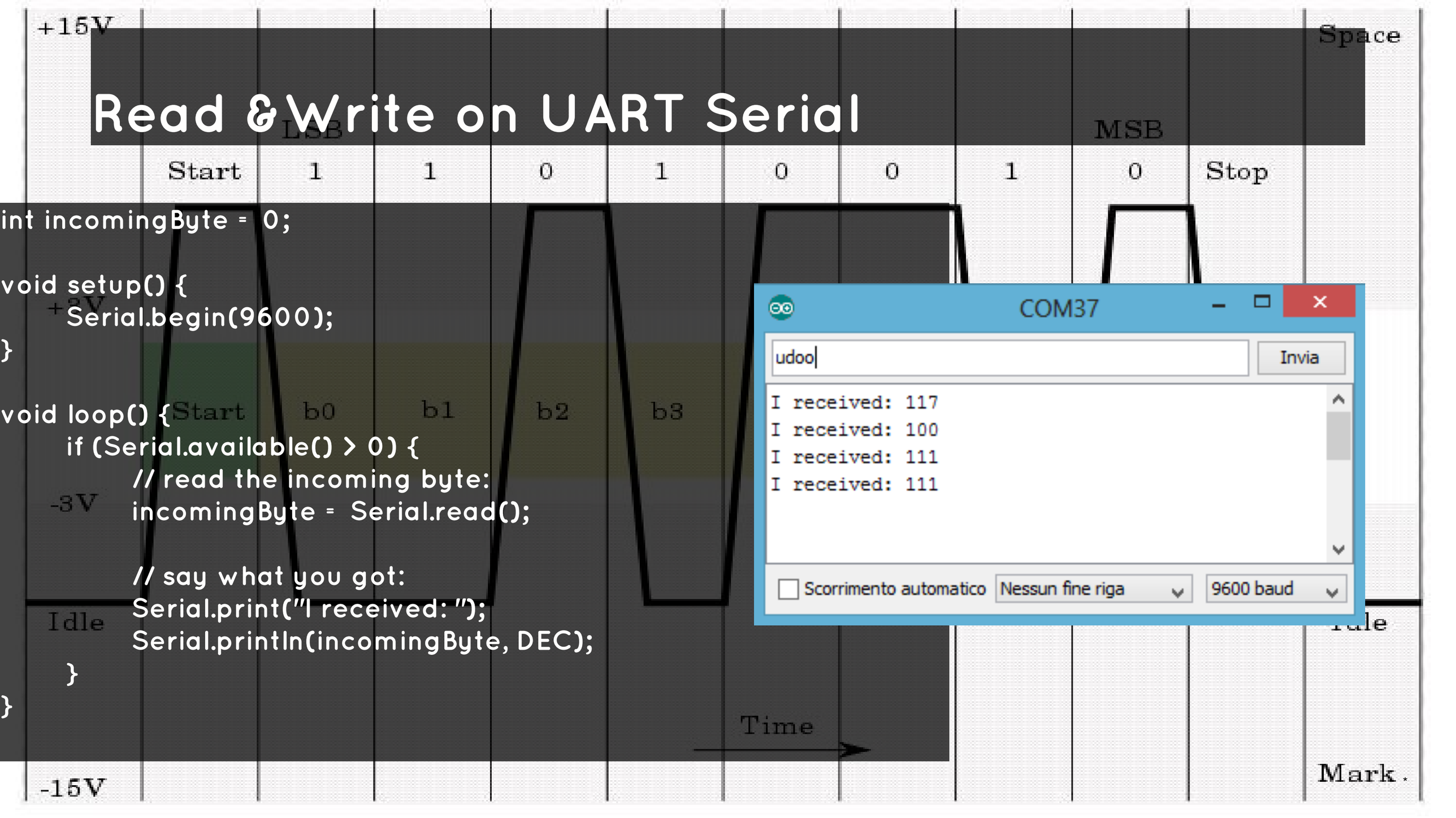
Read &Write on UART Serial

```
int incomingByte = 0;

void setup() {
    Serial.begin(9600);
}

void loop() {
    if (Serial.available() > 0) {
        // read the incoming byte:
        incomingByte = Serial.read();

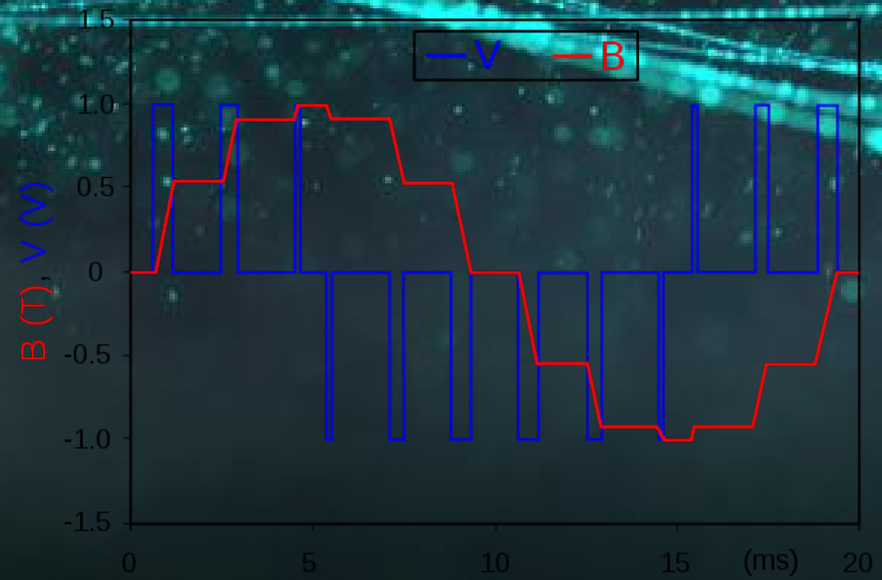
        // say what you got:
        Serial.print("I received: ");
        Serial.println(incomingByte, DEC);
    }
}
```



Read & Write analog signals

Arduino can read voltages between 0 and 3.3 volts. The input range and resolution can be changed using analogReference(). It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.

It can control 2 DAC (Digital to Analog) channel and 12 PWM outputs




```
int analogPin = 3;
int val = 0;

void setup(){
 Serial.begin(9600);
}

void loop(){
 val = analogRead(analogPin);
 Serial.println(val);

 val = map(val, 0, 1023, 0, 255);

 analogWrite(9, val);
 analogWrite(DAC0, val);

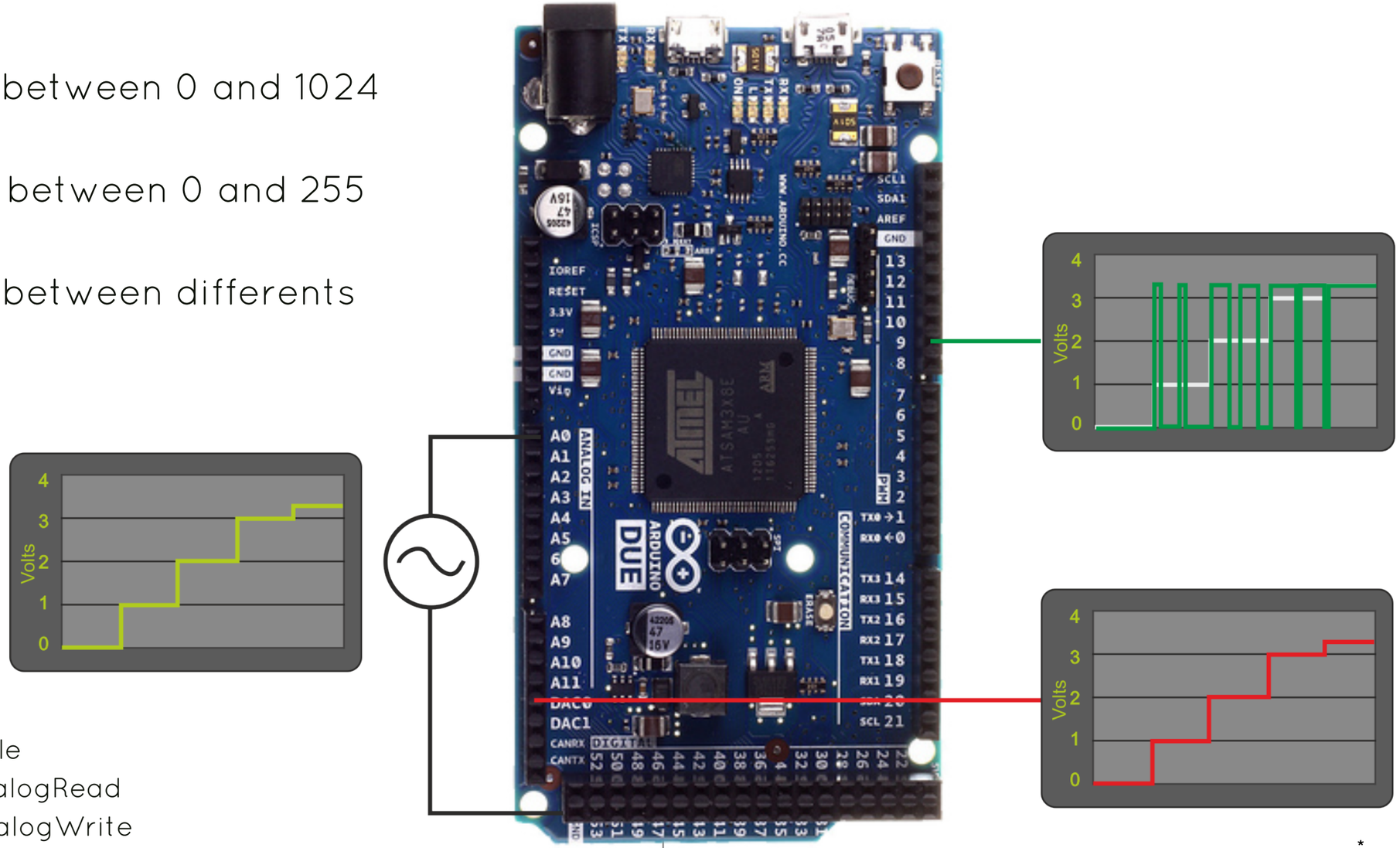
 delay(100);
}
```

Read & Write analog signals

analogRead returns values between 0 and 1024

analogWrite accept values between 0 and 255

map perform a conversion between differents ranges



references

http://arduino.cc/en/Serial/available
http://arduino.cc/en/Reference/analogRead
http://arduino.cc/de/Reference/AnalogWrite

*

Control motors

```
#define DIR_A 12
#define BRAKE_A 9
#define SPEED_A 3

Servo panServo;
int angle;

void setup(){
    Serial.begin(115200);

    pinMode(DIR_A, OUTPUT);
    pinMode(BRAKE_A, OUTPUT);
    pinMode(SPEED_A, OUTPUT);

    panServo.attach(5);

    delay(200);
}
```

```
void loop(){
        digitalWrite(DIR_A, LOW);
        digitalWrite(BRAKE_A, LOW);
        analogWrite(SPEED_A, 200);
        delay(3000);

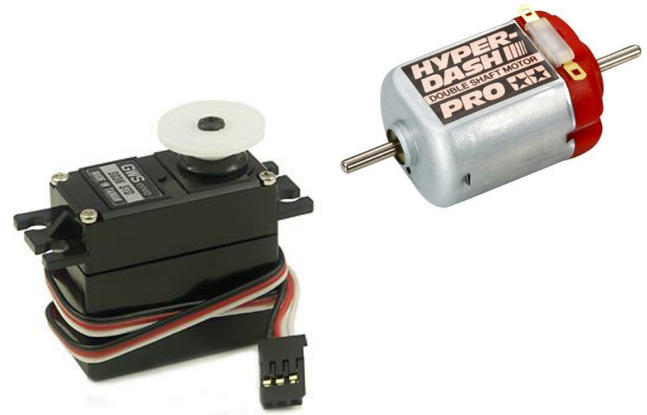
        digitalWrite(BRAKE_A, HIGH);
        analogWrite(SPEED_A, 0);
        delay(3000);

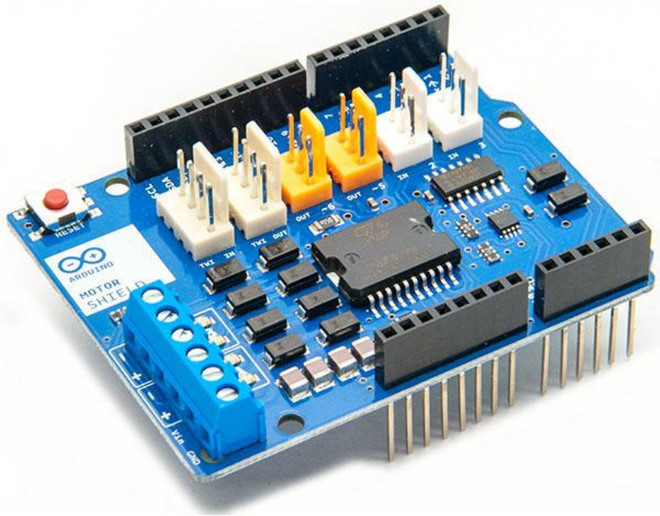
        panServo.write(0);
        panServo.write(180);
        panServo.write(90);

        angle = panServo.read();

        delay(100);
}
```

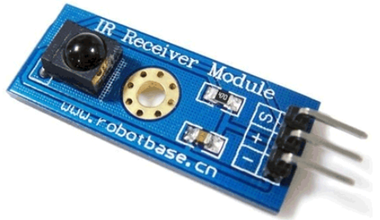
Ingredients:





Read Sensor Data

Ingredients:









Arduino + Wi-Fi + Lan + SD



Arduino + Computer







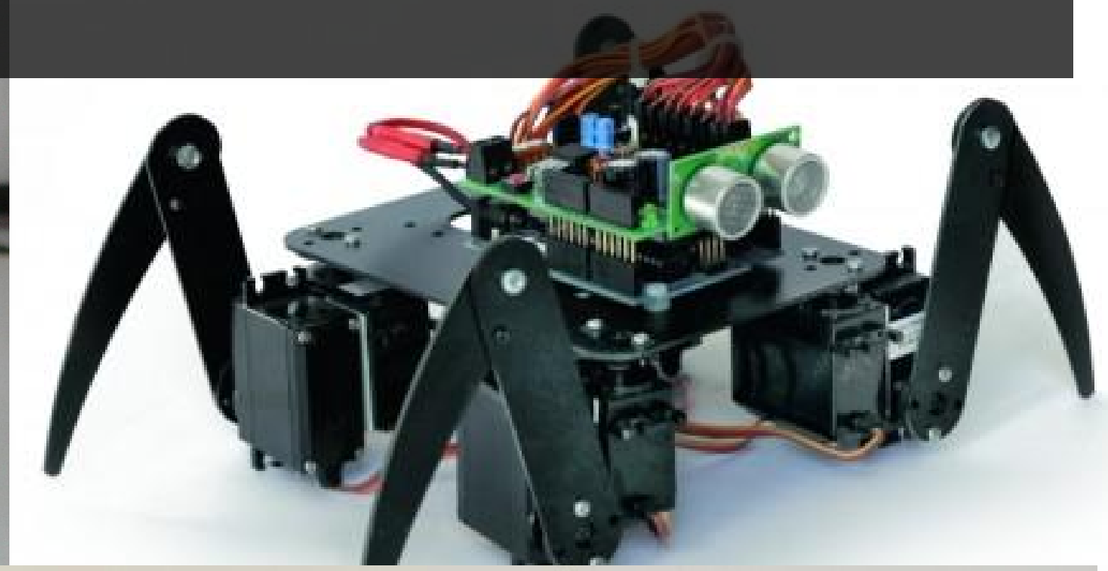
OpenCV

Pd

*

Arduino + Computer







*

Arduino + Computer



Arduino + Android

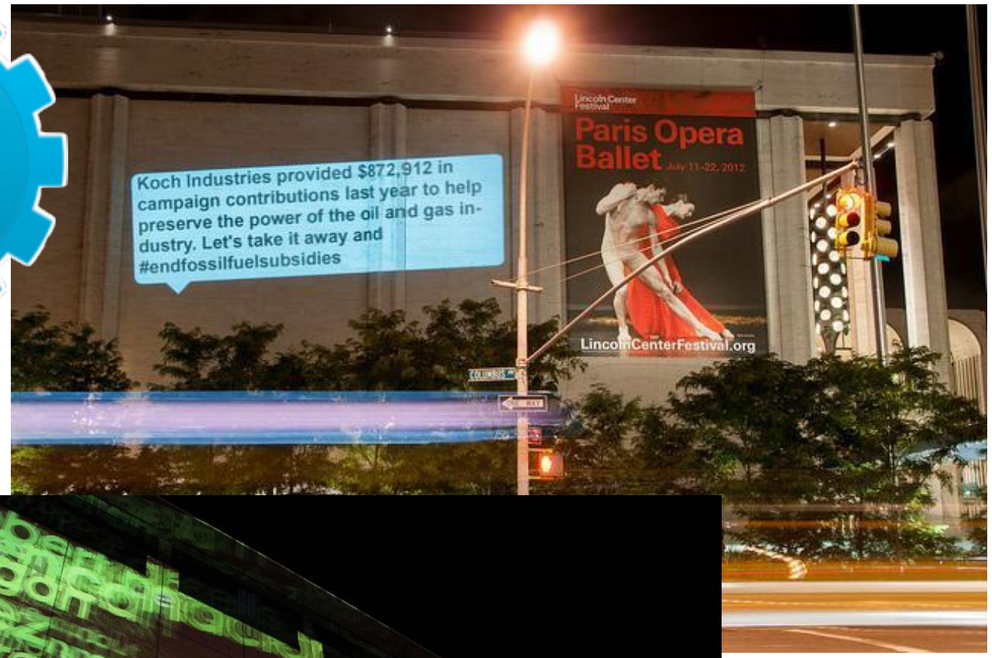






*

Arduino + Web 2.0 services

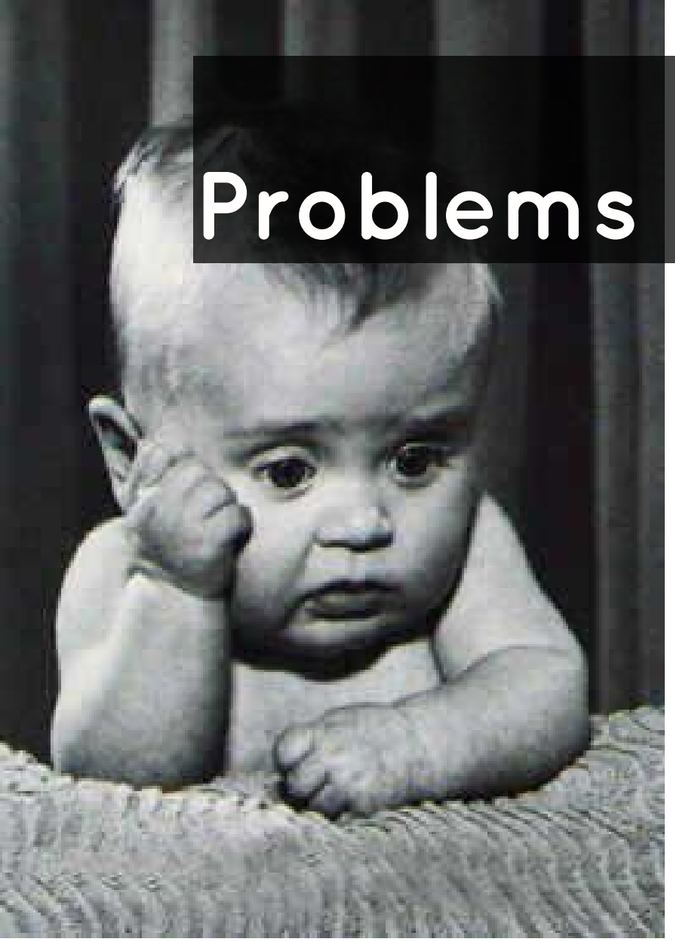




Spotify®







Problems

- For complex project, requiring multimedia, we need a PC coupled with Arduino
- It means USB cable, two power plug
- A PC could be expensive for easy tasks
- A PC could be heavy
- A PC could be “over-sized”





Solution



4x Raspberry Pi

+



Arduino DUE

=



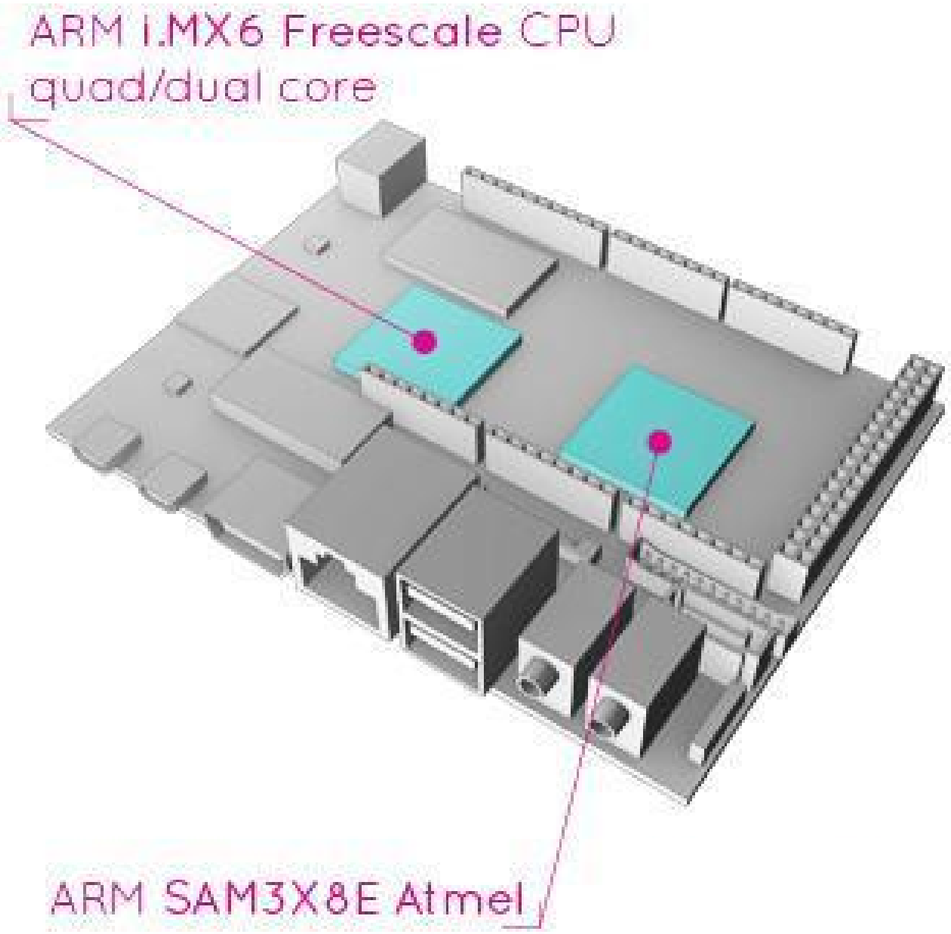
Getting Started

Tutorials

Community

All-in-one solution

For the first time, a **powerful ARM cortex-A9 CPU** for heavy tasks with great performance, and a dedicated **ARM processor for the GPIO** are brought together in one single board computer.



Open Source

UDOO is a proper open source project where both hardware designs and software are open source.

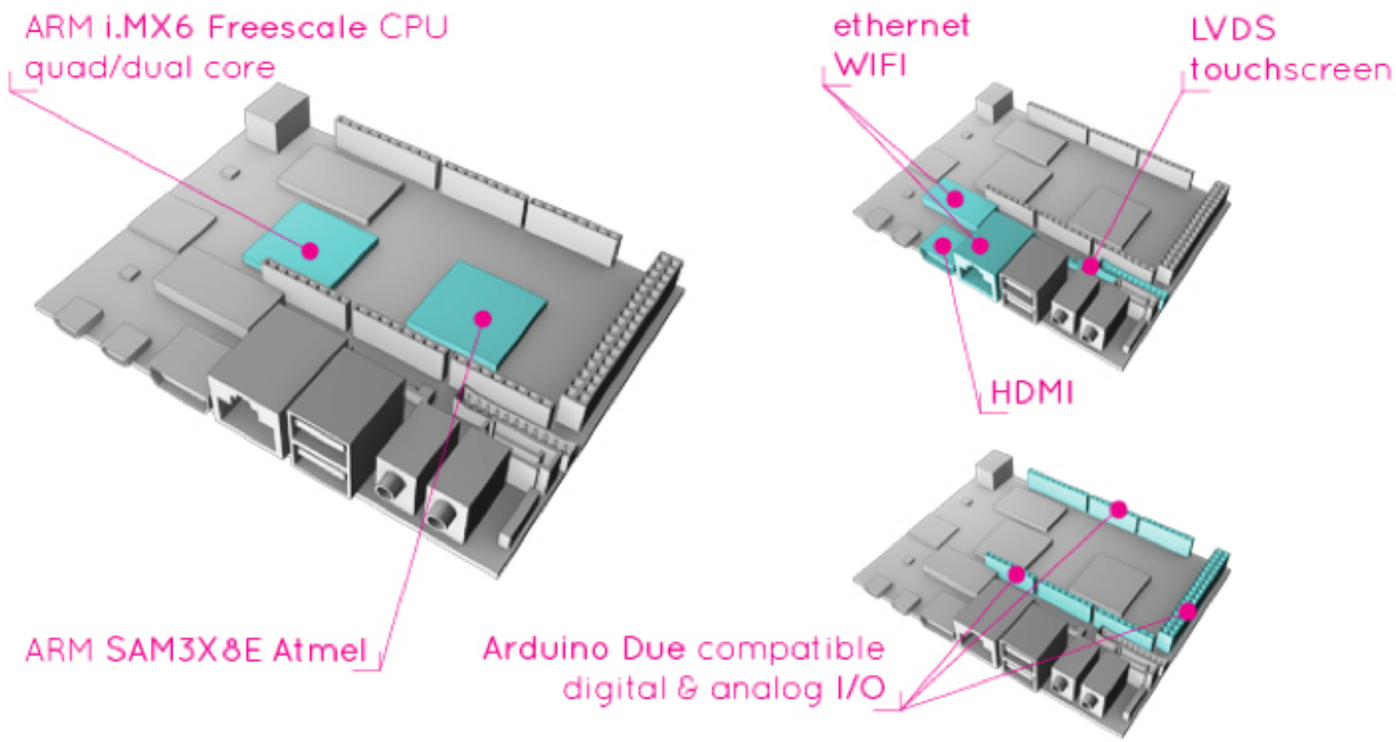
Schematics, 2D/3D drawings, Kernels and U-Boot sources are available to the community in order to improve and extend UDOO capabilities and functionalities.



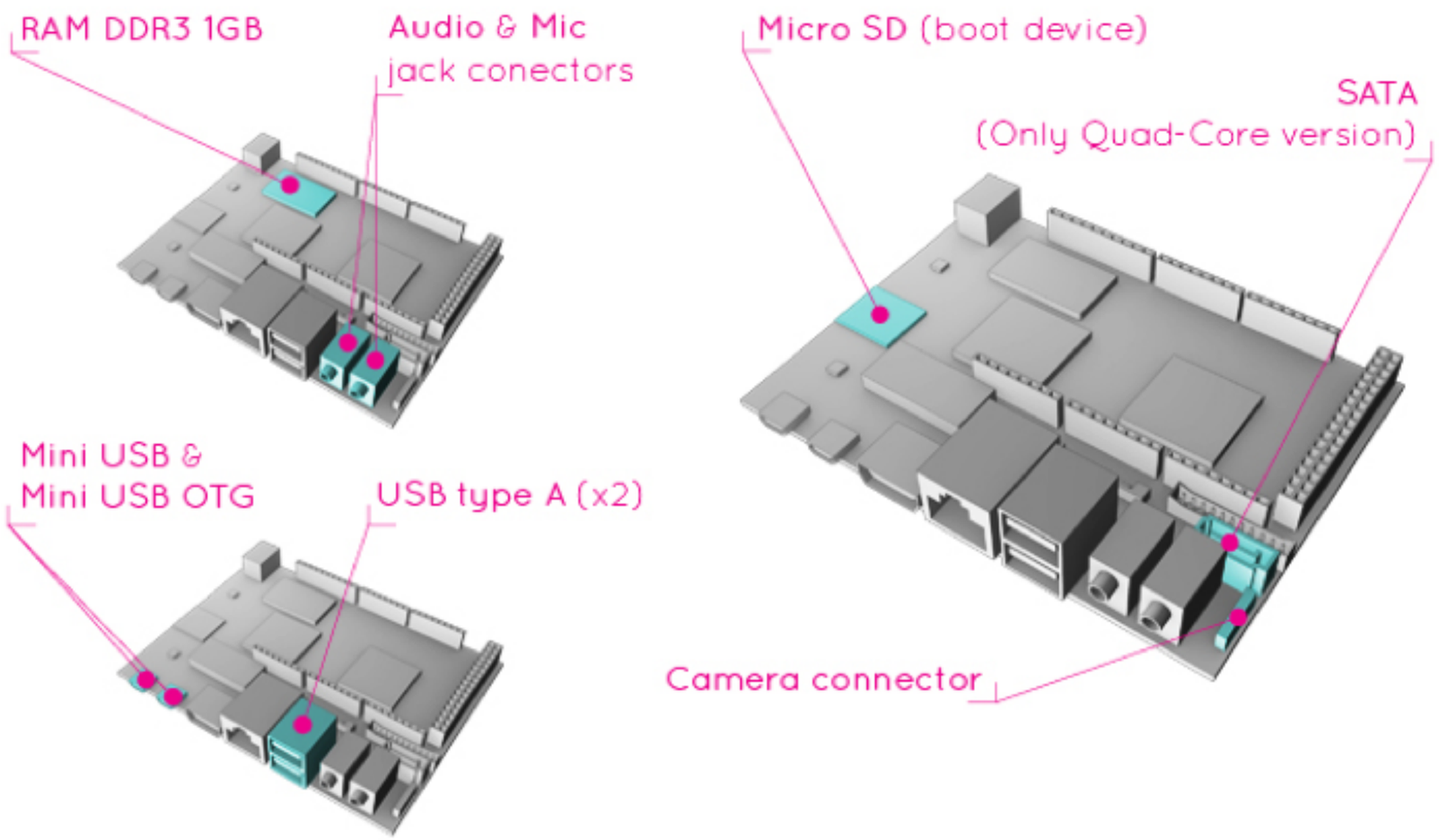
Power and flexibility for any kind of projects

UDOO is a very powerful board based on dual or quad core ARM cortex-A9 CPU with great performance both on Android and Linux OS, and a dedicated ARM processor for the GPIO. These are the main specifications:

- Freescale i.MX 6 ARM Cortex-A9 CPU Dual/Quad core 1GHz
- Integrated graphics, each processor provides 3 separated accelerators for 2D, OpenGL® ES2.0 3D and OpenVG™
- Atmel SAM3X8E ARM Cortex-M3 CPU (same as Arduino Due)
- RAM DDR3 1GB
- 76 fully available GPIO
- Arduino-compatible R3 1.0 pinout
- HDMI and LVDS + Touch (I2C signals)



Power and flexibility for any kind of projects



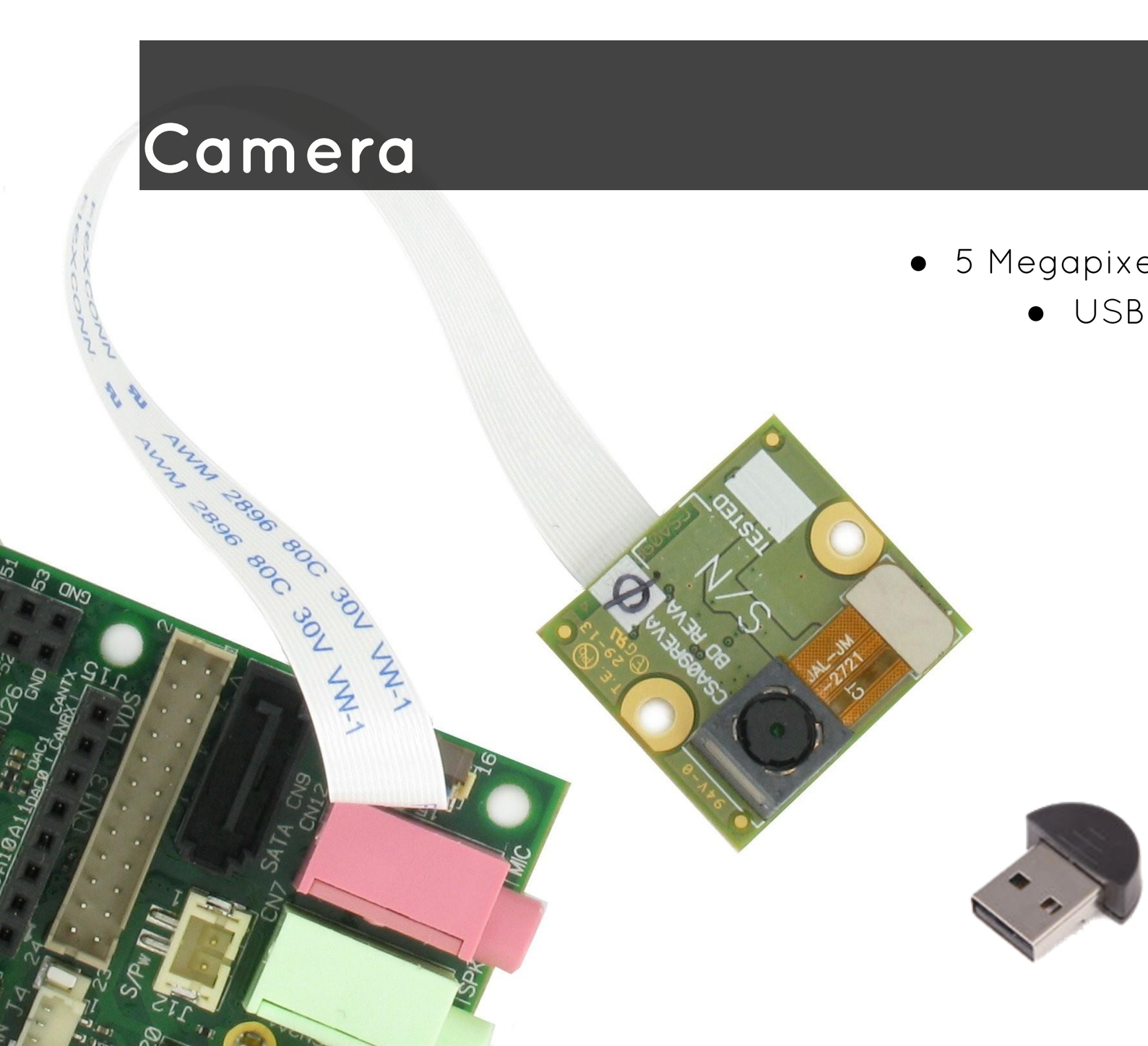
- Ethernet RJ45 (10/100/1000 MBit)
- WiFi Module
- Mini USB and Mini USB OTG
- USB type A (x2) and USB connector (requires a specific wire)
- Analog Audio and Mic
- SATA (Only Quad-Core version)
- Camera connection
- Micro SD (boot device)
- Power Supply 12V and External Battery connector

LVDS & Touchscreens



www.udoo.org

Camera



- 5 Megapixel camera (CSI port)
    - USB Bluetooth 4 module

*

Linux and Android distribution on UDOO



www.udoo.org

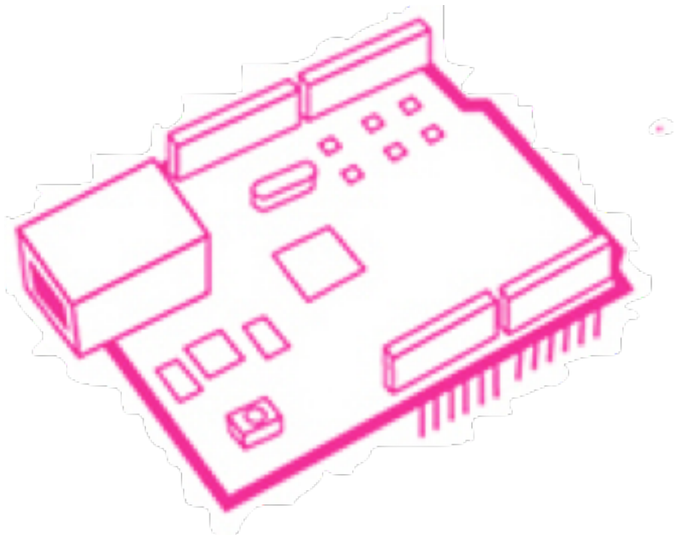
Sofrware on UDOO



www.udoo.org

Arduino-compatible

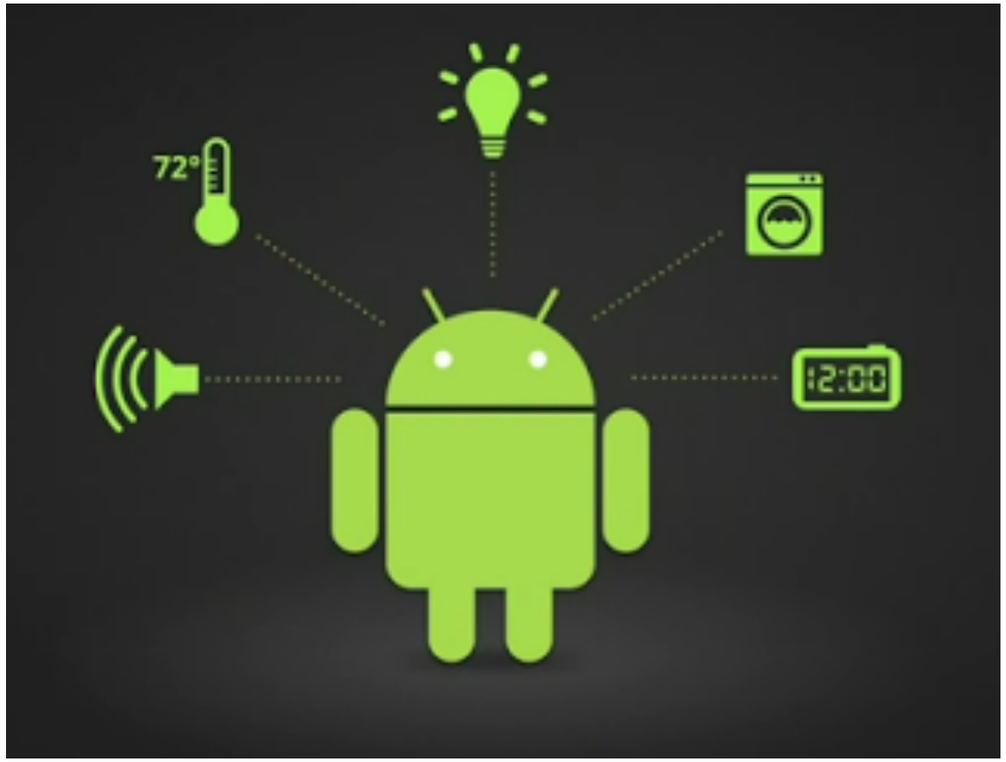
UDOO is compatible with all the sketches, tutorials and resources available on the Arduino community as well as all the shields, sensors and actuators for Arduino DUE available on the market.



Android

Android **4.3 Jelly Beans** runs smoothly on UDOO giving you all the features of an Android device.

Apps interface with Arduino-compatible embedded board through **Accessory Development Kit (ADK)** connection for building accessories and smart devices based on Android.





Processors

Freescale i.Mx6

UDOO is compatible with all the sketches, tutorials and resources available on the Arduino community as well as all the shields, sensors and actuators for Arduino DUE available on the market.

Red indicates change from column to the left

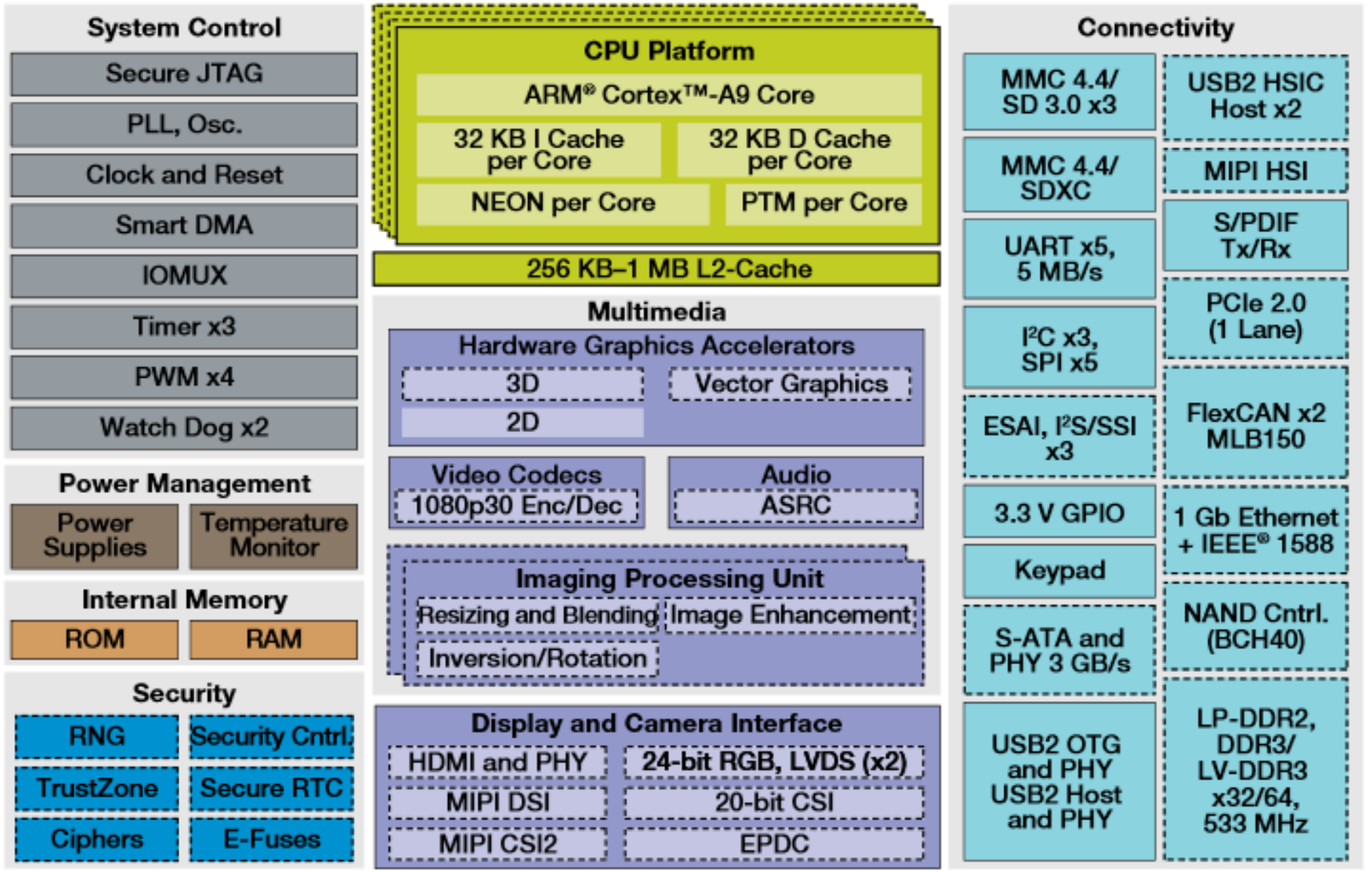
i.MX6SoloLite	i.MX6Solo	i.MX6DualLite	i.MX6Dual	i.MX6Quad
• Single ARM® Cortex™-A9 up to 1.0 GHz	• Single ARM Cortex-A9 up to 1.0 GHz	• Dual ARM Cortex-A9 up to 1.0 GHz	• Dual ARM Cortex-A9 up to 1.2 GHz	• Quad ARM Cortex-A9 up to 1.2 GHz
• 256 KB L2 cache, Neon, VFPvd16 Trustzone	• 512 KB L2 cache, Neon, VFPvd16 Trustzone	• 512 KB L2 cache, Neon, VFPvd16 Trustzone	• 1 MB L2 cache, Neon, VFPvd16 Trustzone	• 1 MB L2 cache, Neon, VFPvd16 Trustzone
• 2D graphics	• 3D graphics with one shader	• 3D graphics with one shader	• 3D graphics with four shaders	• 3D graphics with four shaders
• 32-bit DDR3 and LPDDR2 at 400 MHz	• 2D graphics	• 2D graphics	• Two 2D graphics engines	• Two 2D graphics engines
• Integrated EPD controller	• 32-bit DDR3 and LPDDR2 at 400 MHz	• 64-bit DDR3 and 2-channel 32-bit LPDDR2 at 400 MHz	• 64-bit DDR3 and 2-channel 32-bit LPDDR2 at 533 MHz	• 64-bit DDR3 and 2-channel 32-bit LPDDR2 at 533 MHz
	• Integrated EPD controller	• Integrated EPD controller	• Integrated SATA-II	• Integrated SATA-II
	• HDMIv1.4 controller plus PHY	• HDMIv1.4 controller plus PHY	• HDMIv1.4 controller plus PHY	• HDMIv1.4 controller plus PHY
	• LVDS controller plus PHY	• LVDS controller plus PHY	• LVDS controller plus PHY	• LVDS controller plus PHY
	• PCIe controller plus PHY	• PCIe controller plus PHY	• PCIe controller plus PHY	• PCIe controller plus PHY
	• MLB and FlexCan controllers	• MLB and FlexCan controllers	• MLB and FlexCan controllers	• MLB and FlexCan controllers

www.udoo.org

Freescale i.Mx6

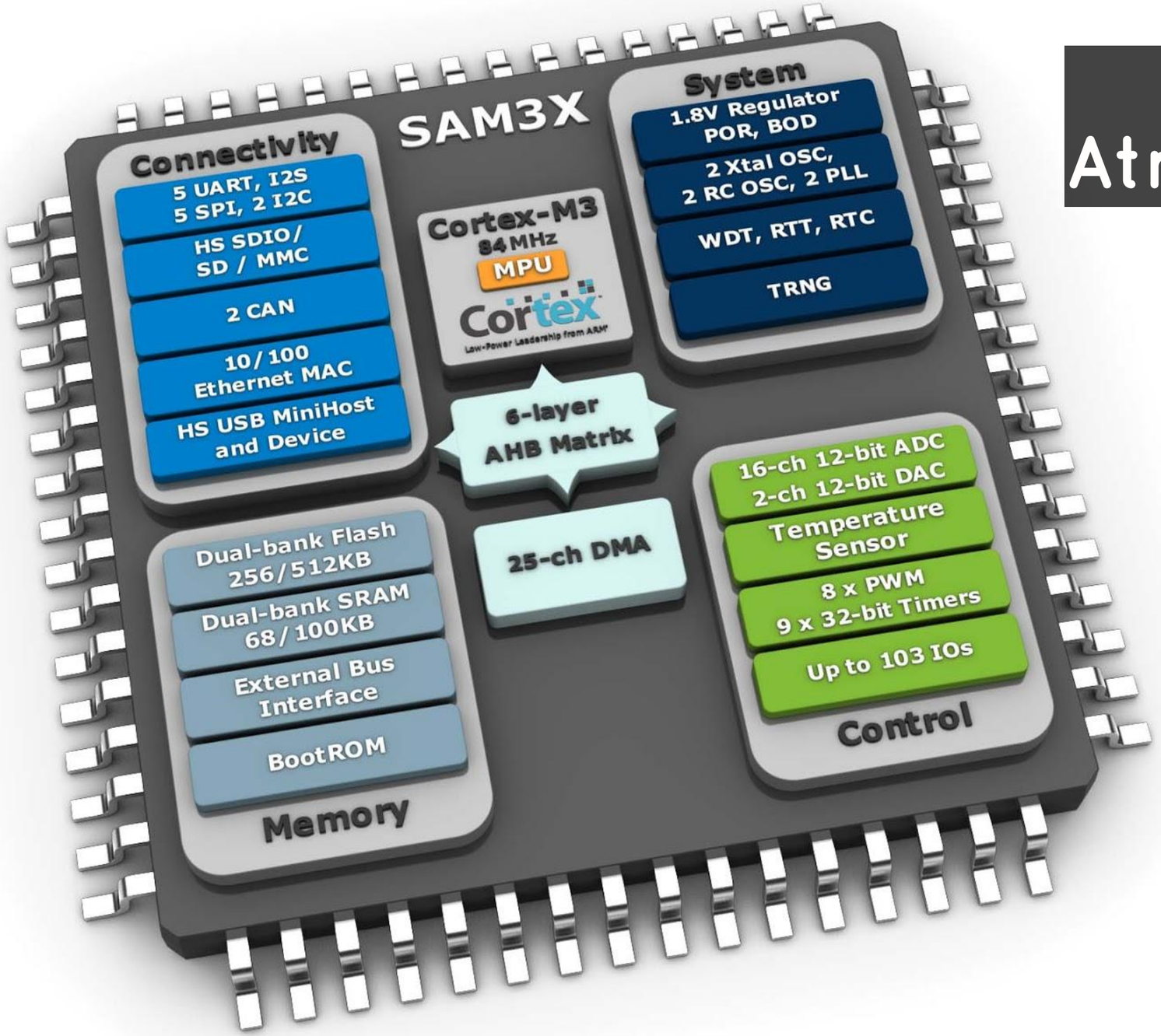
Exported Features:

- SPI
- I2C
- UART
- SPDIF
- AUDMUX
- CAN
- GPIOs



www.udoo.org

Atmel SAM3X


SAM3X
Connectivity
5 UART, I2S
5 SPI, 2 I2C
HS SDIO/
SD / MMC
2 CAN
10/100
Ethernet MAC
HS USB MiniHost
and Device
Cortex-M3
84 MHz
MPU
Cortex
Low-Power Leadership from ARM
System
1.8V Regulator
POR, BOD
2 Xtal OSC,
2 RC OSC, 2 PLL
WDT, RTT, RTC
TRNG
6-layer
AHB Matrix
25-ch DMA
Dual-bank Flash
256/512KB
Dual-bank SRAM
68/100KB
External Bus
Interface
BootROM
Memory
16-ch 12-bit ADC
2-ch 12-bit DAC
Temperature
Sensor
8 x PWM
9 x 32-bit Timers
Up to 103 IOs
Control

*

CPU muxing







Register Rename

Register Alias History Tables (2x126)
Register Alias Tables
uOp Queue
Micro code Sequencer
Micro code ROM & Flash
Trace Cache
Fill Buffers
Distributed Tag comparators
24 bit virtual Tags

Instruction Queue (for less critical fields of the uOps)

General Instruction Address Queue & Memory Instruction Address Queue (queues register entries and latency fields of the uOps for scheduling)

Floating Point, MMX, SSE2 Renamed Register File 128 entries of 128 bit.

uOp Schedulers

FP Move Scheduler: (8x8 dependency matrix)

Parallel (Matrix) Scheduler for the two double pumped ALU's

General Floating Point and Slow Integer Scheduler: (8x8 dependency matrix)

Load / Store uOp Scheduler: (8x8 dependency matrix)

Load / Store Linear Address Collision History Table

Integer Execution Core

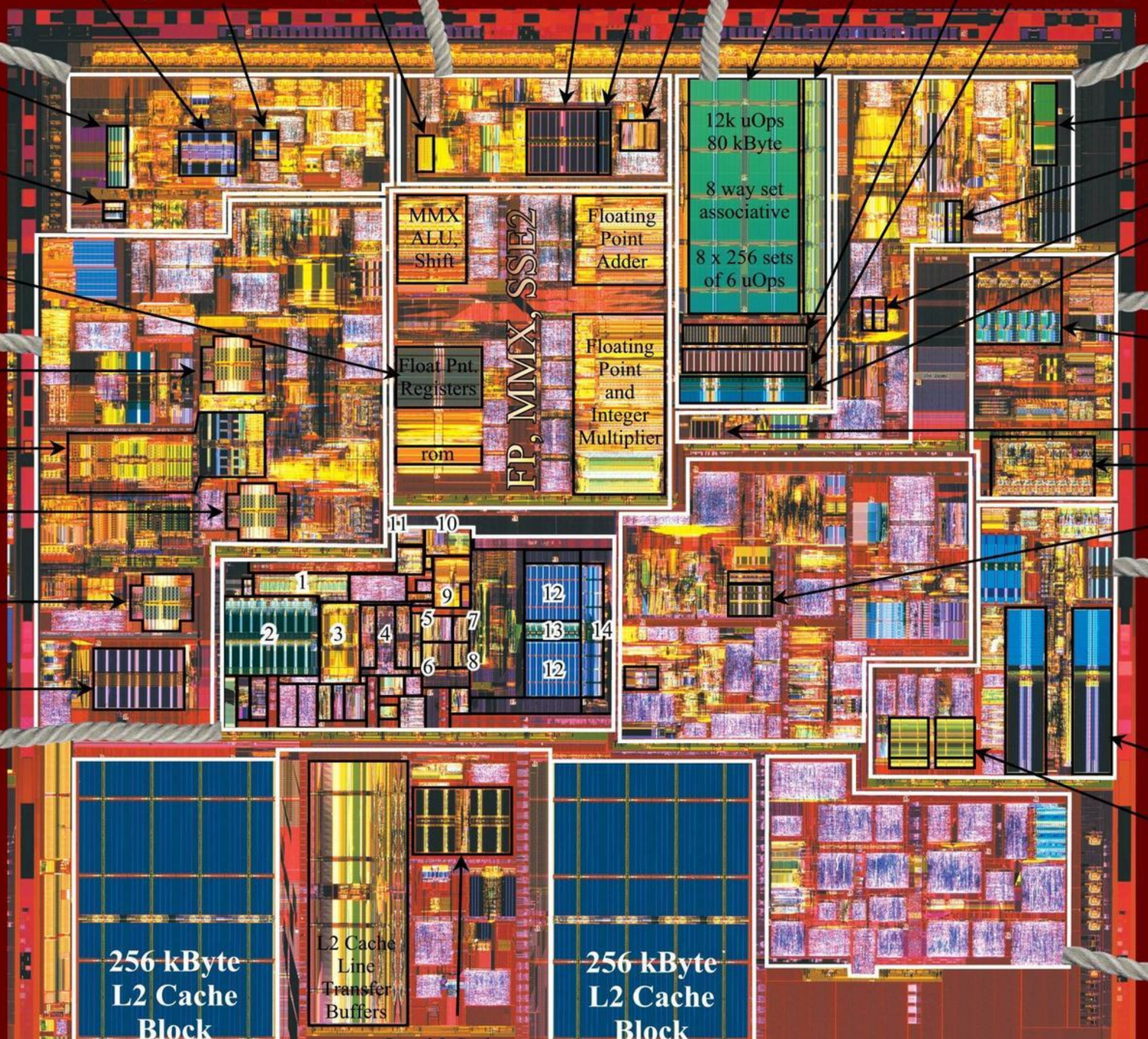
(1) uOp Dispatch unit & Replay Buffer
Dispatches up to 6 uOps / cycle

(2) Integer Renamed Register File
128 entries of 32 bit + 6 status flags
12 read ports and six write ports

(3) Databus switch & Bypasses to and from the Integer Register File.

(4) Flags, Write Back

(5) Double Pumped ALU 0



Trace Cache Access, next Address Predict

Trace Cache Branch Prediction Table (BTB), 512 entries.

Return Stacks (2x16 entries)

Trace Cache next IP's (2x)

Miscellaneous Tag Data

Instruction Decoder

Up to 4 decoded uOps/cycle out. (from max. one x86 instr/cycle)
Instructions with more than four are handled by Micro Sequencer

Trace Cache LRU bits

Raw Instruction Bytes in

Data TLB, 64 entry fully associative, between threads dual ported (for loads and stores)

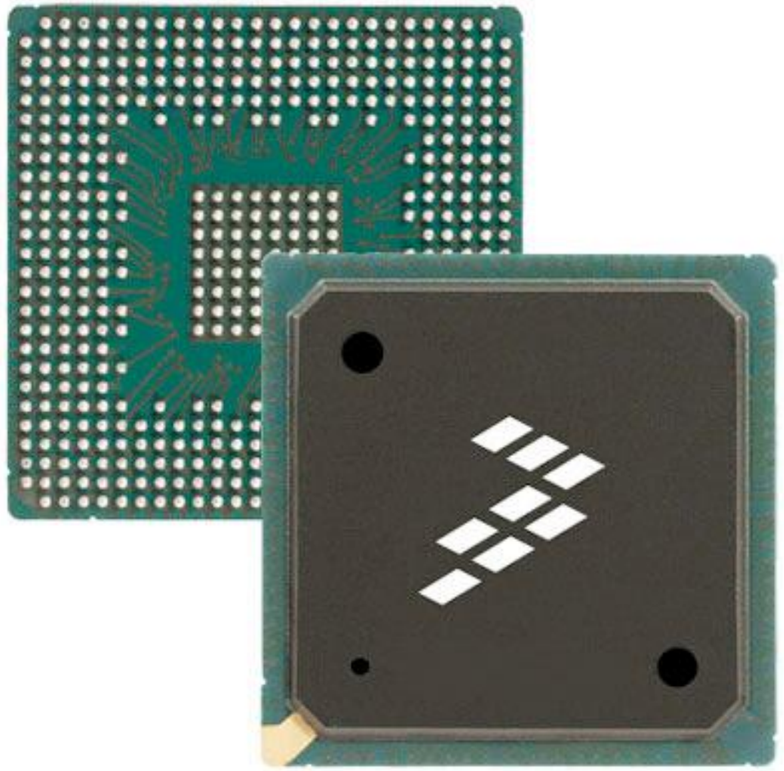
Instruction Fetch from L2 cache and Branch Prediction

Front End Branch Prediction Tables (BTB), shared, 4096 entries in total

Instruction TLB's 2x64 entry, fully associative for 4k and 4M pages. In: Virtual address [31:12] Out: Physical address [35:12] + 2 page level bits

Pinmuxing & GPIO

Physical Pins



SD2_DAT2	ALT0	SD2_DATA2	HYS
	ALT1	ECSPI5_SS1	PUS
	ALT2	EIM_CS3	PUE
	ALT3	AUD4_TXD	PKE
	ALT4	KEY_ROW6	ODE
	ALT5	GPIO1_IO13	SPE
			DSE
			SRE
SD2_DAT3	ALT0	SD2_DATA3	HYS

Functionalities

www.udoo.org

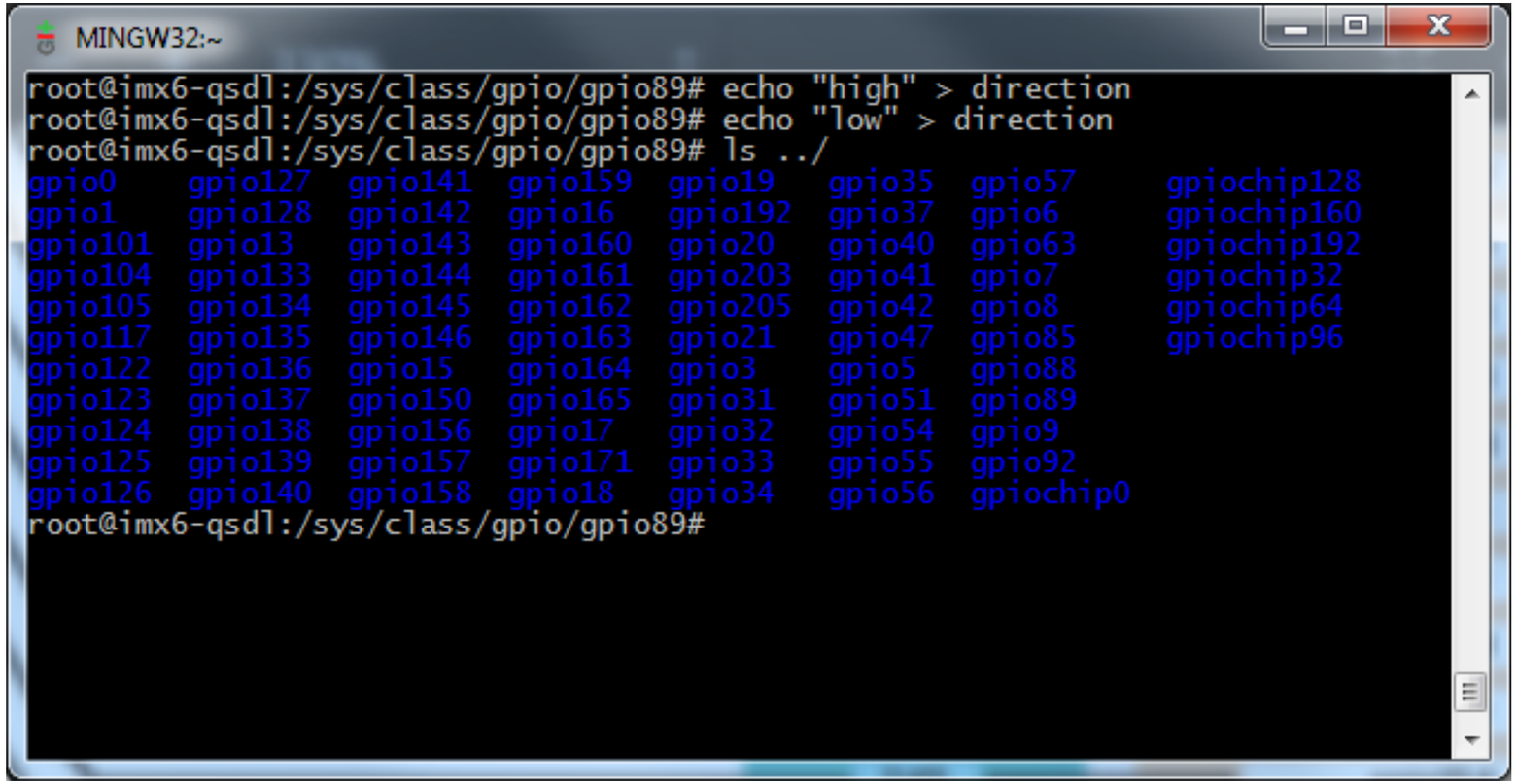
GPIO

Modes:

- Input

read voltage and return LOW or HiGH value

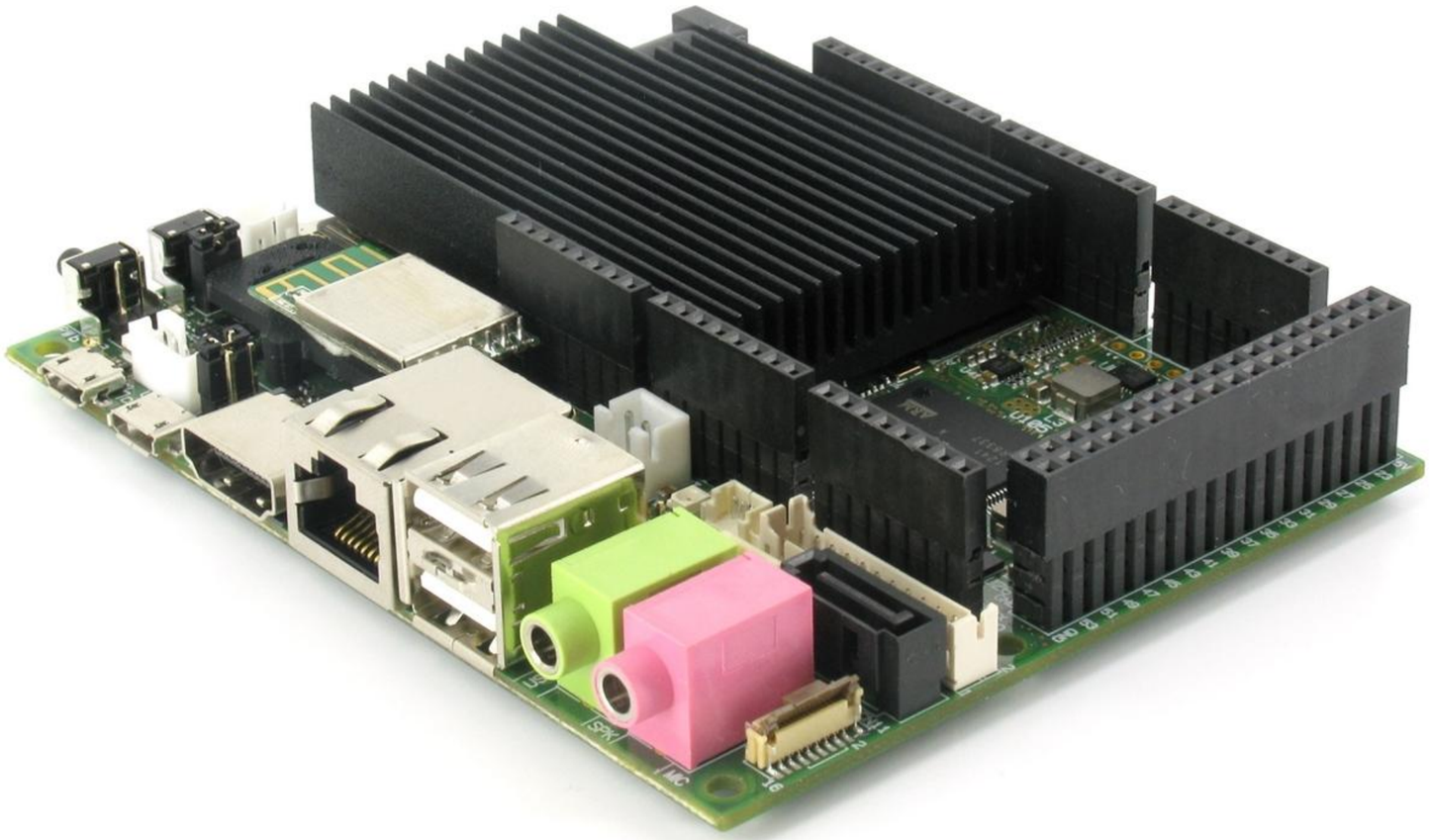
- Output
  - Low
  - High





Communication

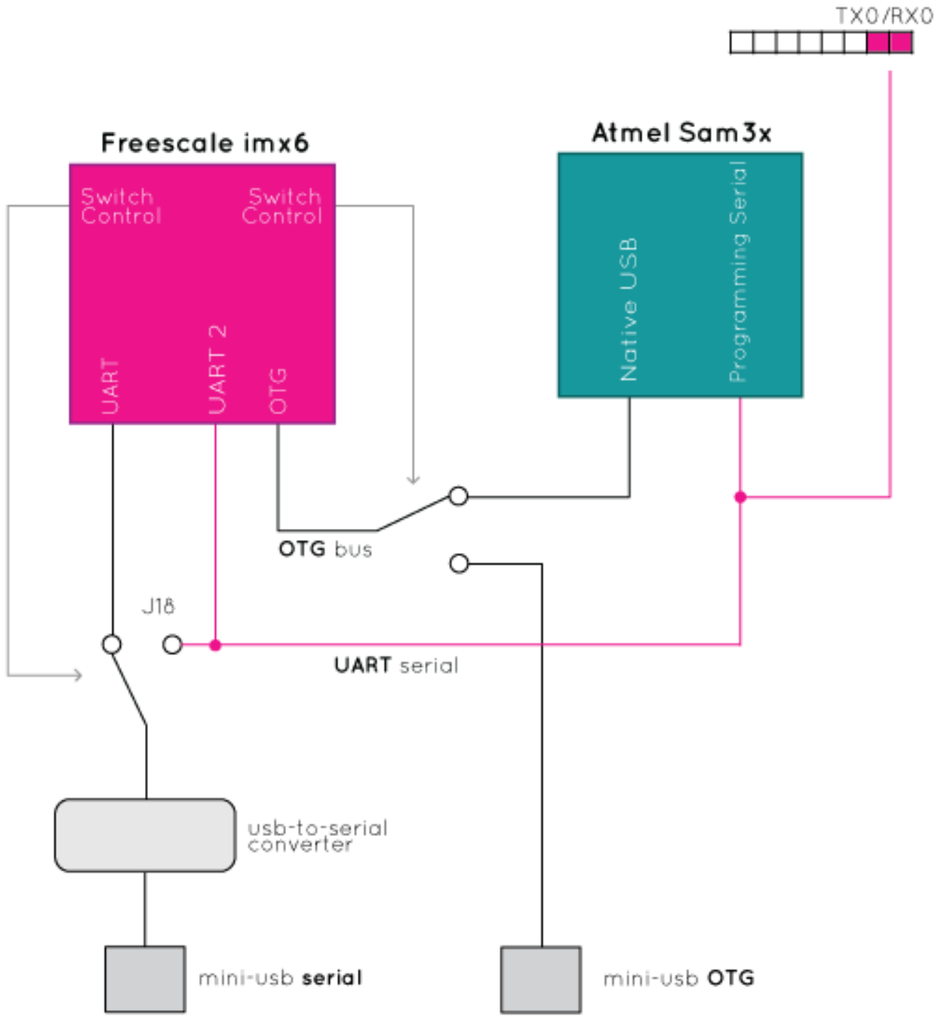
External Communication



- Wireless
- Ethernet
- USB
- GPIOs

www.udoo.org

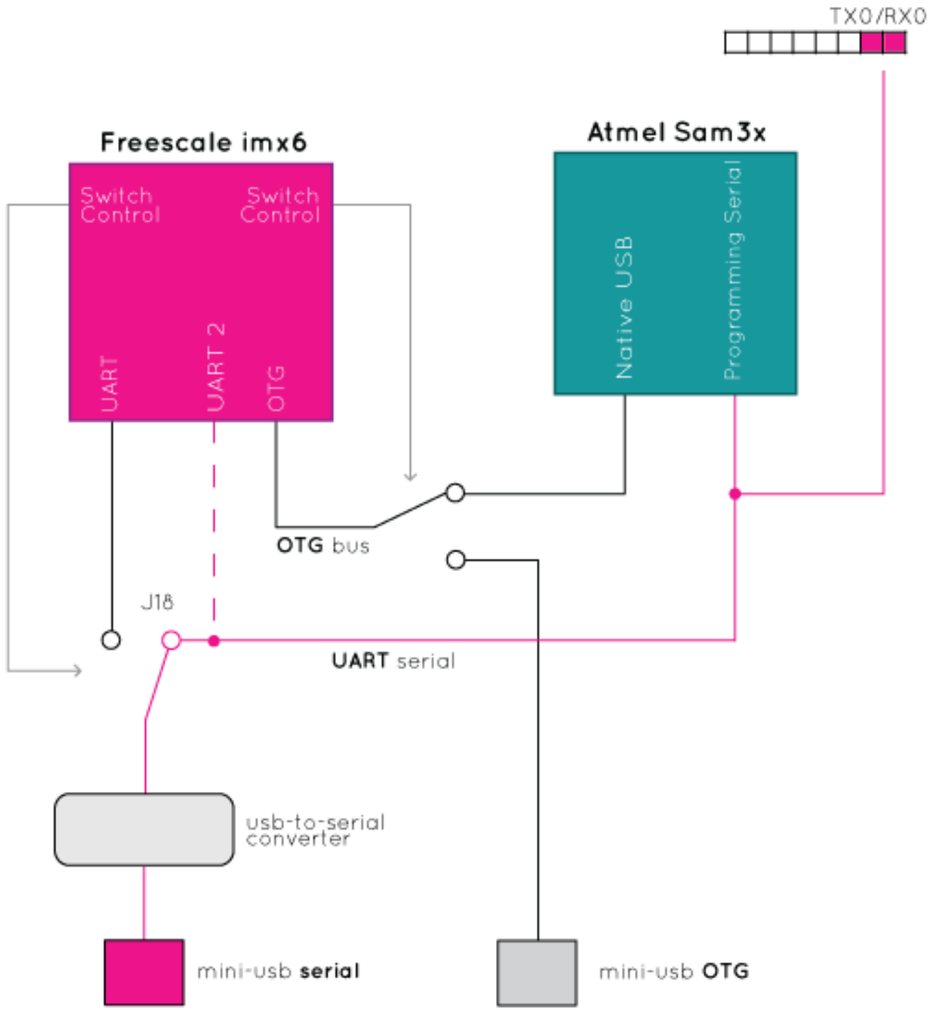
Internal Communication - Serial UART



Communication between iMX6 and SAM3X through UART serial

- Potentially 4 devices connected on the same line
- External microUSB switched by J18
- iMx6 can enable/disable its uart connection, reducing number of device connected

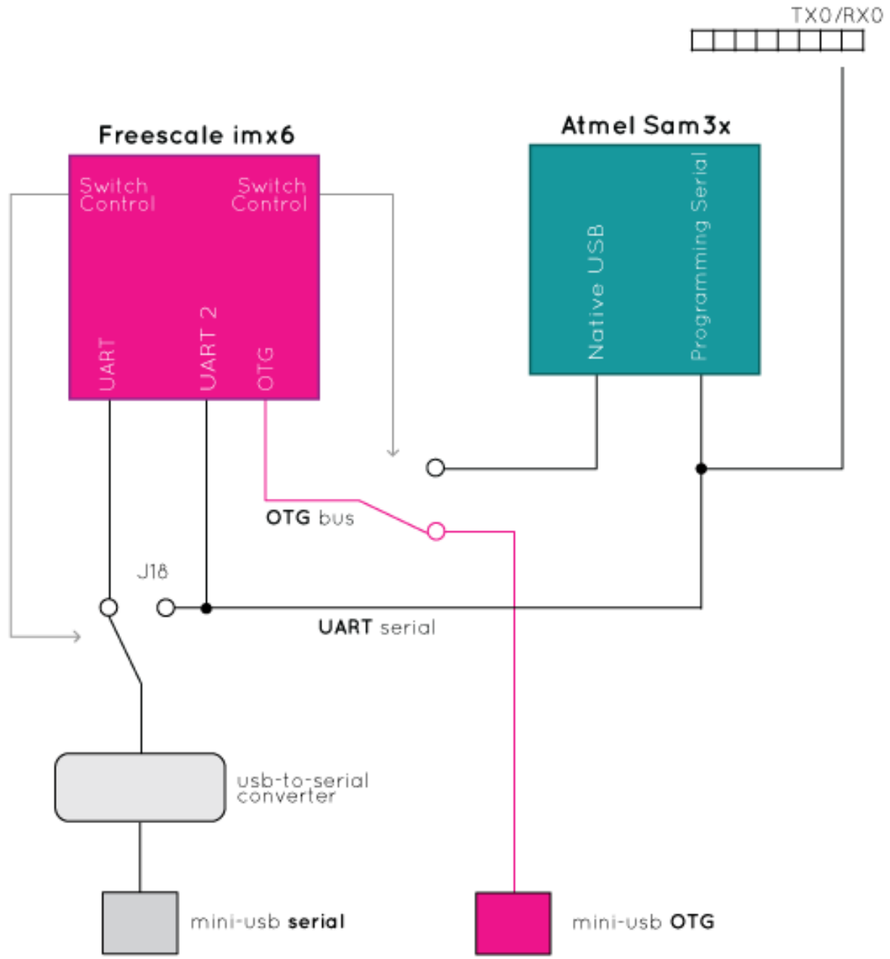
Internal Communication - Serial UART II



Connection to the SAM3X from external computer

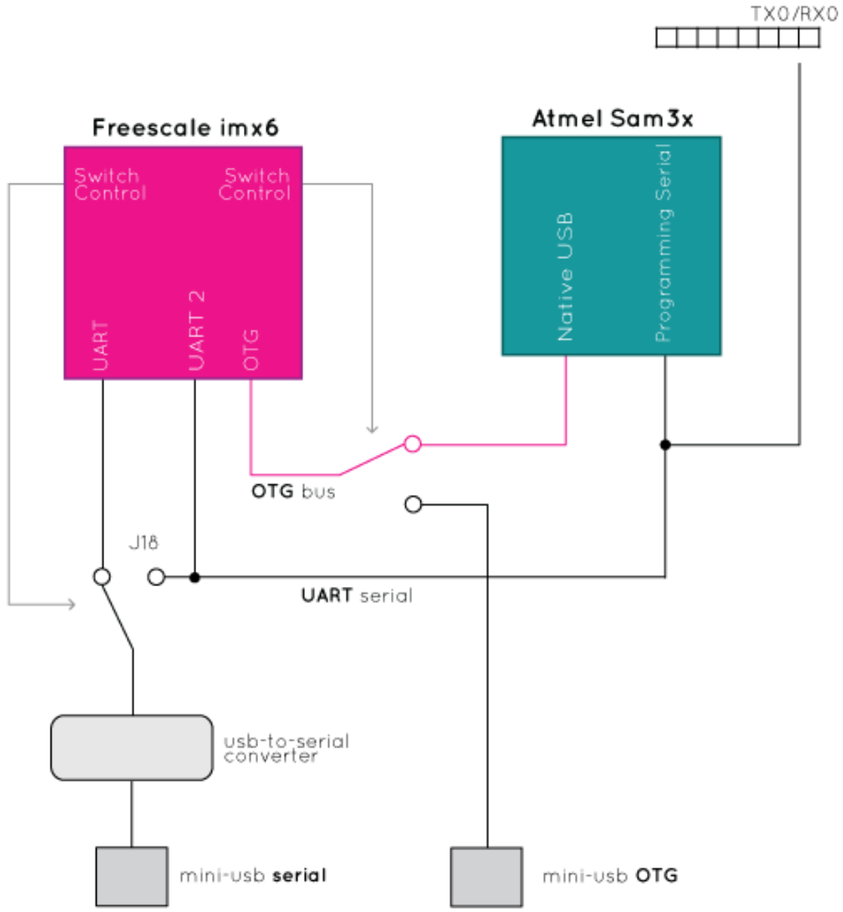
- SAM3X is always connected to UART line.
- Disabling iMx6 connection is possible to program SAM3X from an external pc.
- A kernel module detect when an external pc try to program the microcontroller.

Internal Communication - OTG bus



External OTG connection to IMX6

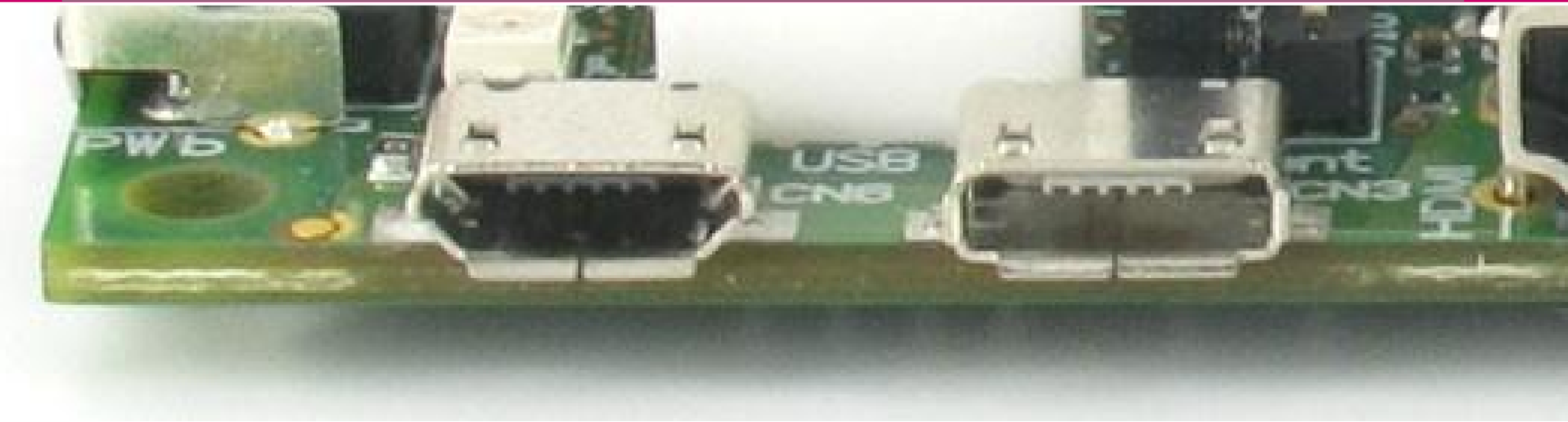
programming



OTG connection between IMX6 and SAM3X

execution

Internal Communication - External uUSB



- iMx6 serial debug console
- shared internal UART (SAM3X)
- Android OTG debug connection (ADB)



BSP overview, create SD and first boot

Board Support Package (BSP)

In embedded systems, a board support package (BSP) is implementation specific support code (software) for a given (device motherboard) board that conforms to a given operating system. It is commonly built with a bootloader that contains the minimal device support to load the operating system and device drivers for all the devices on the board.

Board Support Package (BSP)

BSP is supposed to perform the following operations

- Initialize the processor
- Initialize the bus
- Initialize the interrupt controller
- Initialize the clock
- Initialize the RAM settings
- Configure the segments
- Load and run bootloader from flash

Board Support Package (BSP)

- uboot
- kernel
- file system

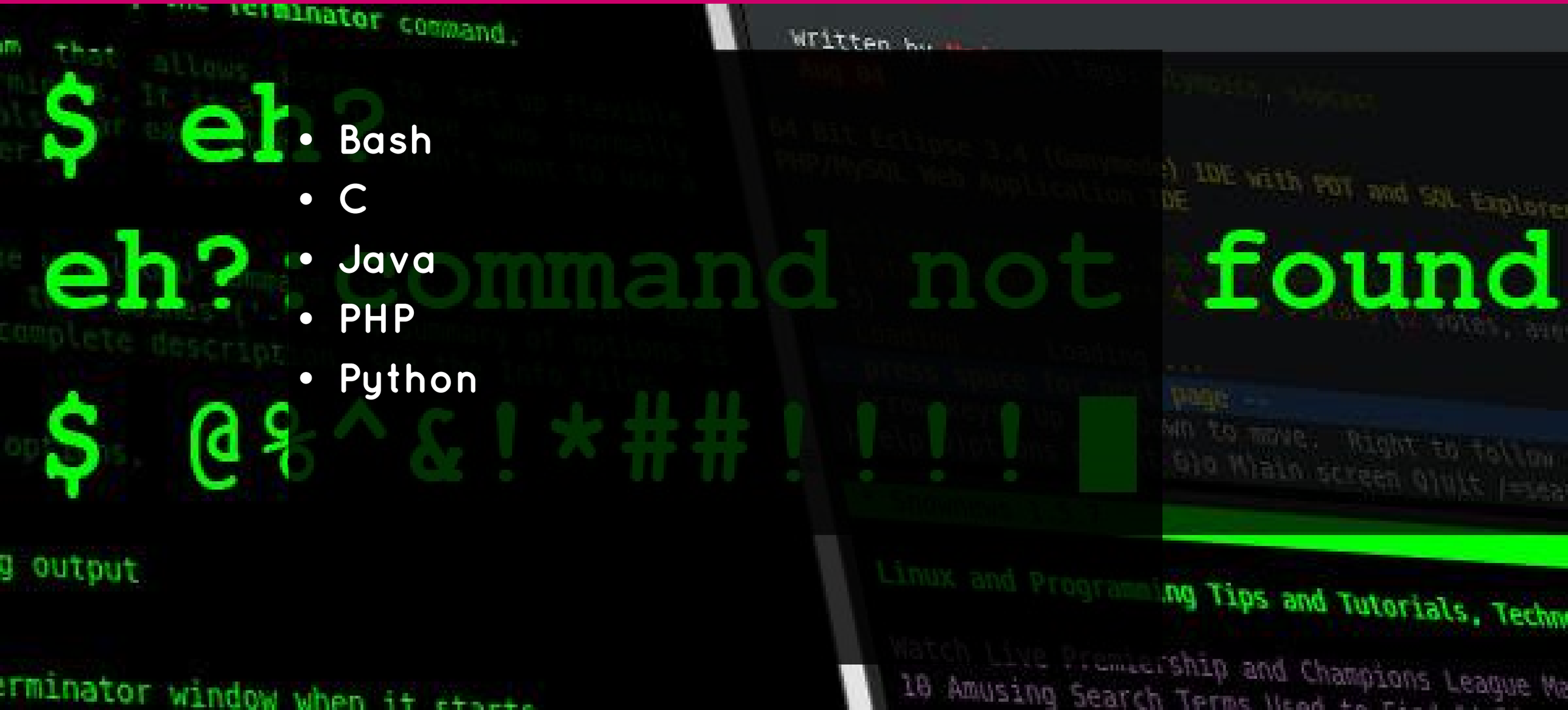






Linux Hello world development

Linux - example languages



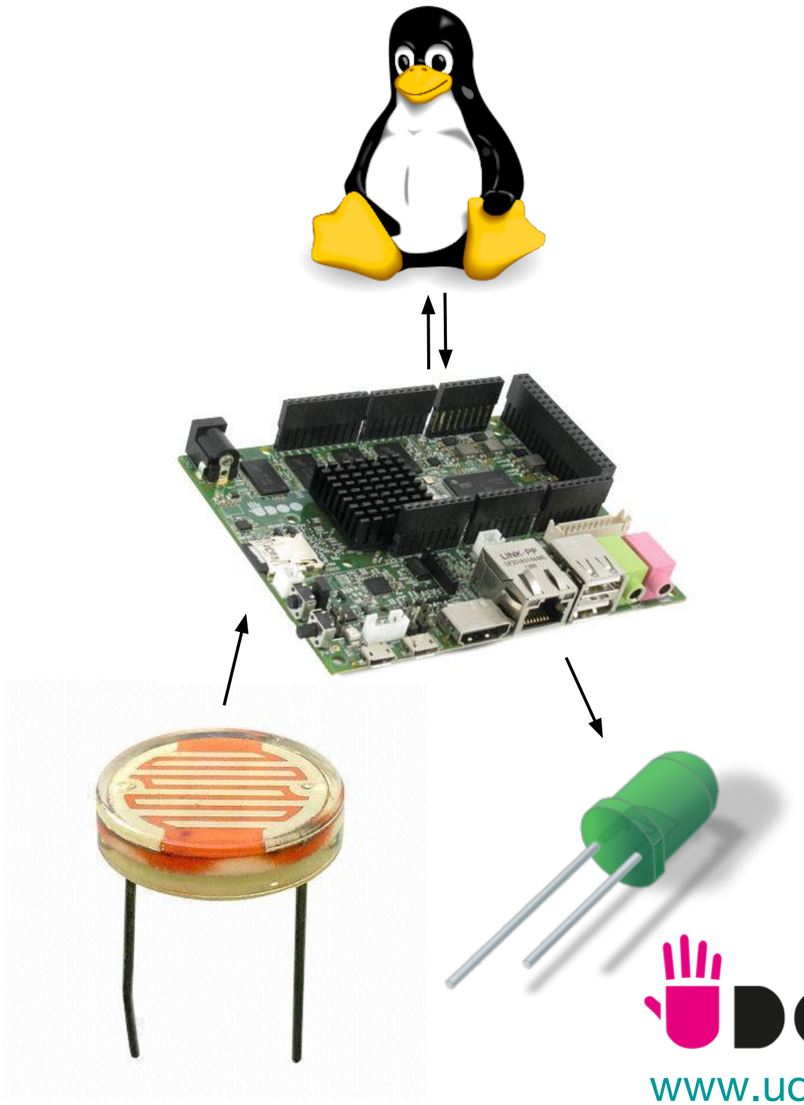
- Bash
- C
- Java
- PHP
- Python

Linux - project description

- Our project will blink a led with defined sequnces.

Linux send an integer between 1 and 9, then the led blink this number of times.

- A light sensor turn on the led if there is no light.
- We control the LED with Linux software in different languages.



Linux - Arduino sketch

- **Initialize Serial communication**
- **Initialize Input and output pins**
- Read light value
- write value on serial
- decide if it's dark
- read serial
- blink led if input is correct

```
int lightIntensity = 0;
int readedValue = 0;

void setup(){

   Serial.begin(9600);
   pinMode(13, OUTPUT);

}

   // ...
```

Linux - Arduino sketch

- Initialize Serial communication
- Initialize Input and output pins
- **Read light value**
- **write value on serial**
- **decide if it's dark**
- read serial
- blink led if input is correct

```
  // ...

void loop(){
  lightIntensity = analogRead(1);

  Serial.println(lightIntensity);

  if(lightIntesity < TH){
     digitalOutput(13, HIGH);
  else
     digitalOutput(13, LOW);

}
```



Linux - Arduino sketch

- Initialize Serial communication
- Initialize Input and output pins
- Read light value
- write value on serial
- decide if it's dark
- **read serial**
- **blink led if input is correct**

```
    // ...
    if (Serial.available() > 0) {

        readedByte =  Serial.read();

        if(readedByte > 47 && readedByte < 58){
            int i =  48;
            for(i =  48; i<58; i++){
                digitalOutput(13, HIGH);
                delay(100);
                digitalOutput(13, LOW);
                delay(100);
            }
        }
    }
} // END LOOP
```

Linux - C ansi

```
char *portname = "/dev/ttyUSB1"

int fd = open (portname, O_RDWR | O_NOCTTY | O_SYNC);
if (fd < 0){
      error_message ("error %d opening %s: %s", errno, portname, strerror (errno));
      return;
}

// set speed to 115,200 bps, 8n1 (no parity)
set_interface_attribs (fd, B115200, 0);  // set no blocking

set_blocking (fd, 0);
// send 7 character greeting
write (fd, argv[0], 1);
char buf [100];

 // read up to 100 characters if ready to read
int n = read (fd, buf, sizeof buf);
```

WE NEED EXTRA LIB

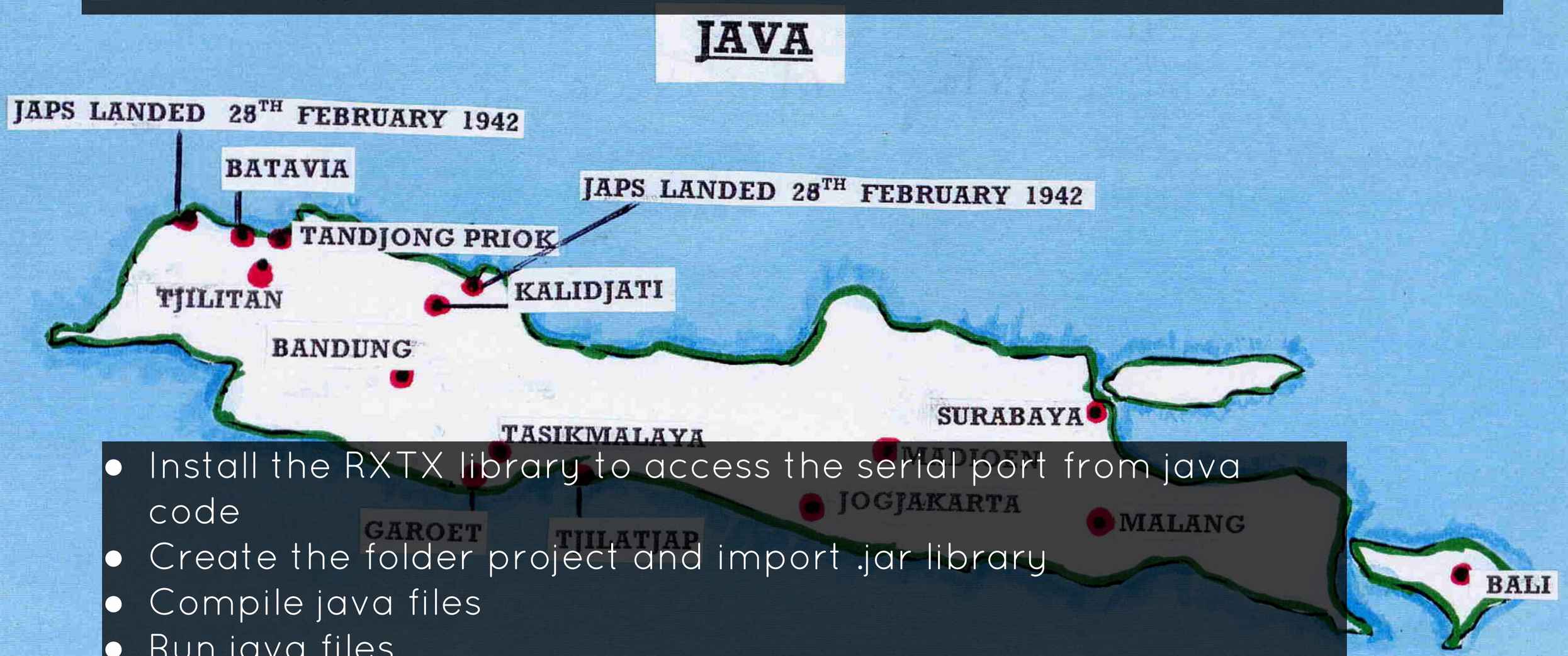
Linux - C ansi



Compile the code using GCC:

```
> gcc serial_test.c -o serial_test
> chmod 777 serial_test
> ./serial_test 3
```

Linux - JAVA



- Install the RXTX library to access the serial port from java code
- Create the folder project and import .jar library
- Compile java files
- Run java files

Linux - PHP

- Install LAMP server

Linux - Python

- Install LAMP server

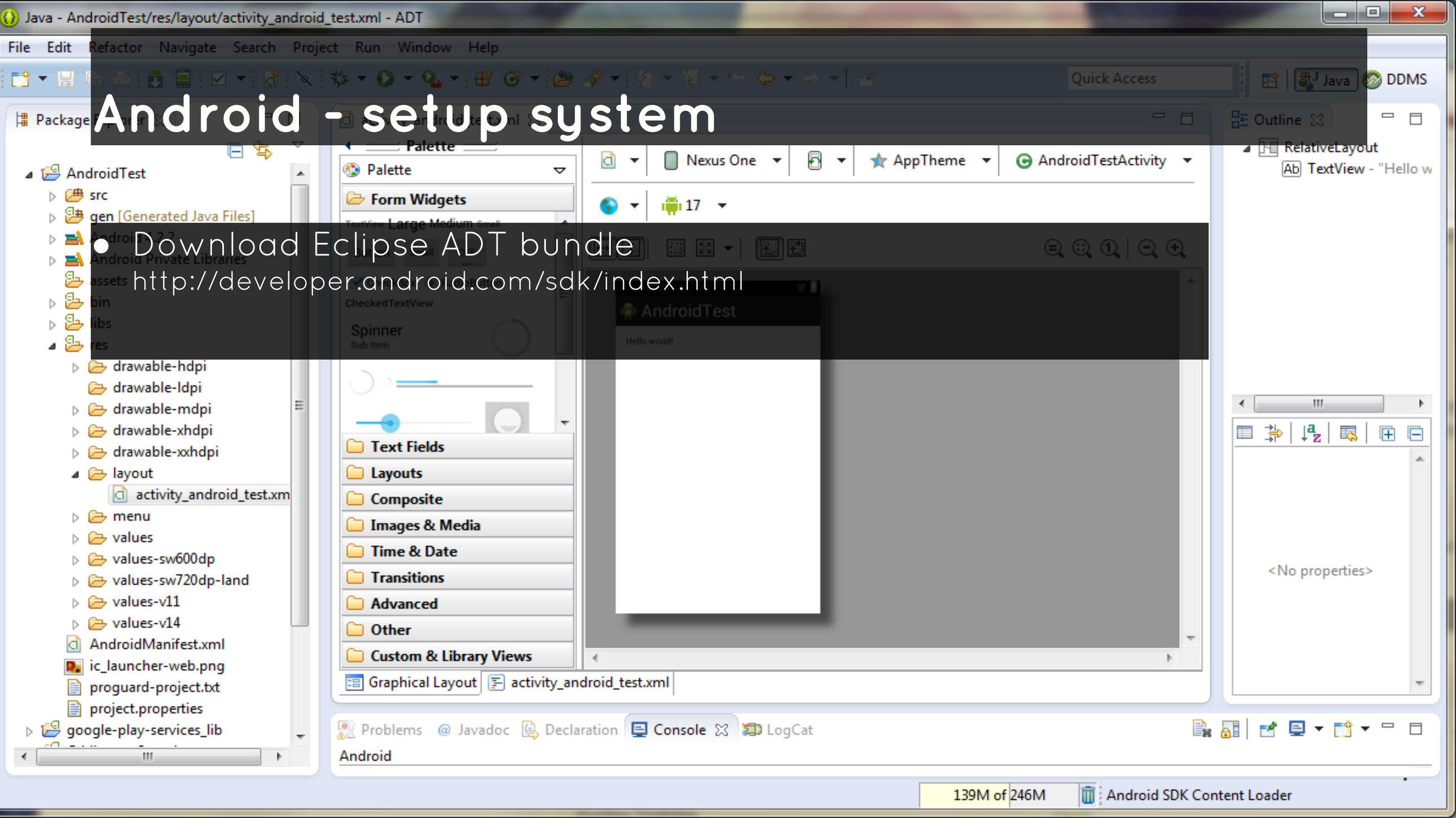




Android Hello world development

Android - setup system

- Download Eclipse ADT bundle
  http://developer.android.com/sdk/index.html



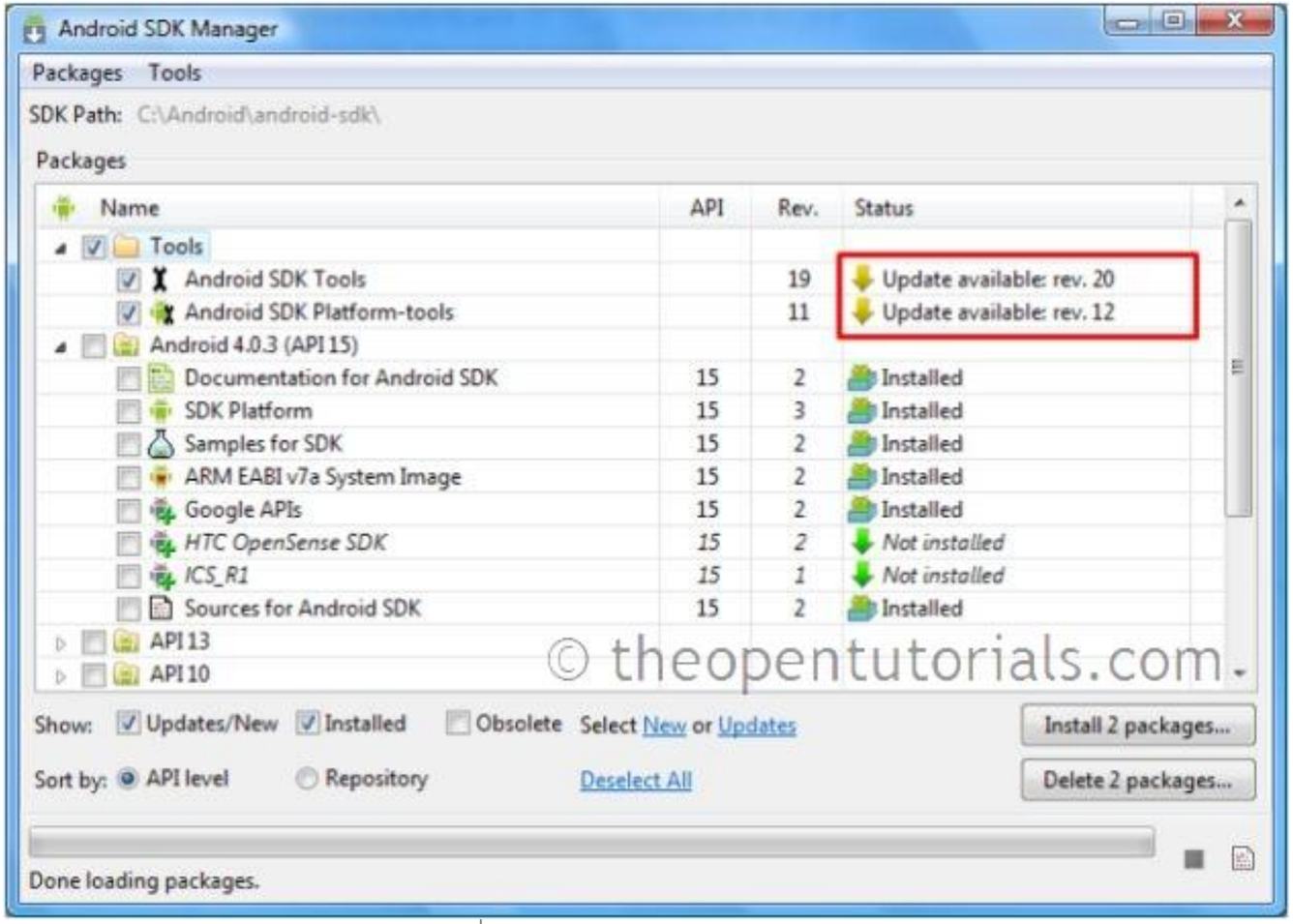
Android - setup system

- Alternatively you can use Android Studio (still in beta)
  http://developer.android.com/sdk/installing/studio.html



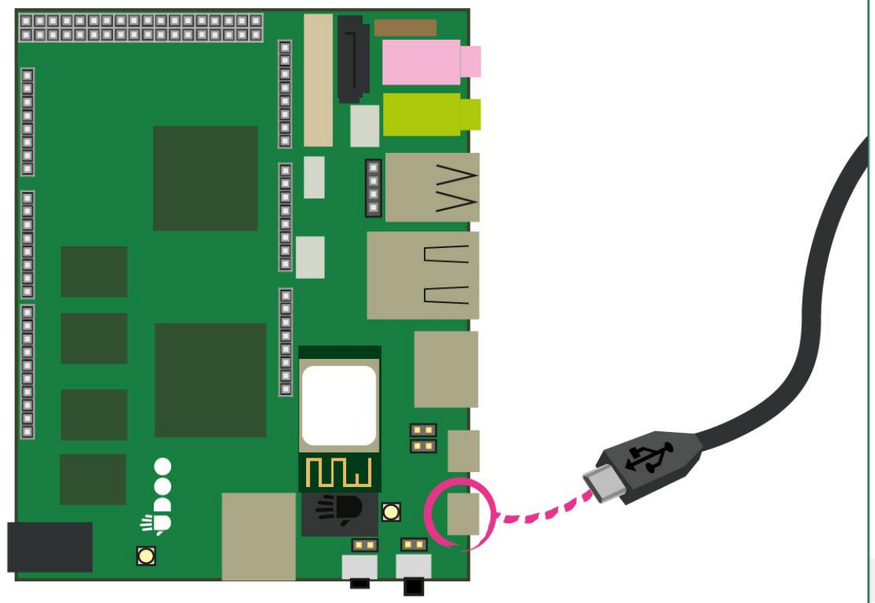
Android - setup system

- Download SDK manager and last Android API (19)



Android - ADK sketch

- Add extra libs
- Create buffer
- Initialize Usb.task
- read and write in buffer
- compile and send



UDOO_adk_default | Arduino 1.5.2

File Modifica Sketch Strumenti Aiuto

UDOO_adk_default §

```
#define RCVSIZE 128

void setup(){
    Serial.begin(115200);
    pinMode(LED_PIN, OUTPUT);
    cpu_irq_enable();
    delay(200);
}


void loop(){
    uint8_t buf[RCVSIZE];
    uint32_t nbread = 0;

    Usb.Task();

    if (adk.isReady()) {
      adk.read(&nbread, RCVSIZE, buf);// read data into buf variabl
      if (nbread > 0) {
        if (buf[0] == 1) // compare received data
          digitalWrite(LED_PIN,HIGH); // turn on light
        else
```

Android - Create ADK project

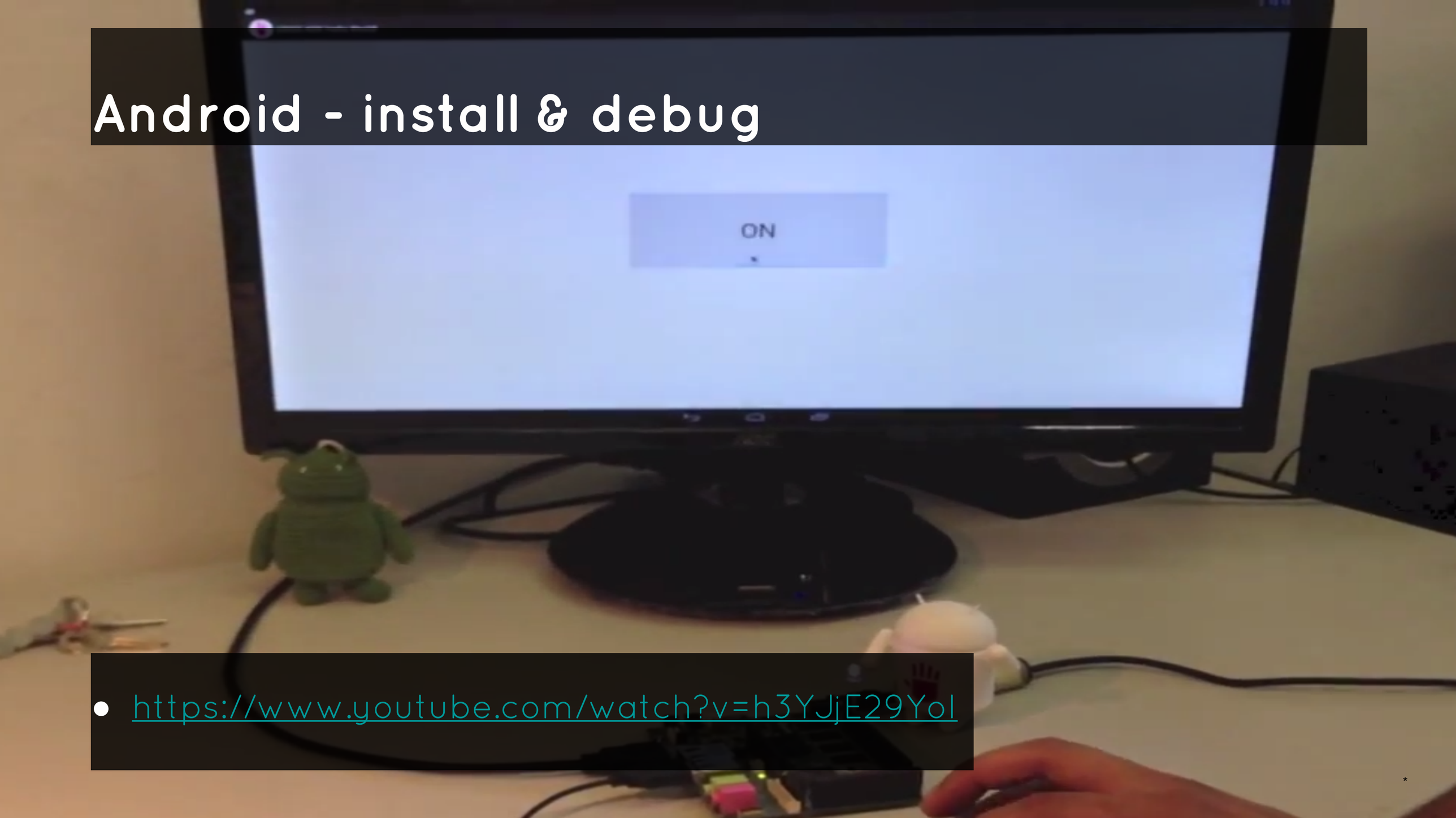
- AndroidManifest.xml
- accessory_filter.xml
- Import methods
- Open input e output streams
- Recieve and send messages

```
if (mAccessory != null) {
    byte[] buffer = new byte[1];

    buffer[0] = (byte) byteToSend;

    if (mOutputStream != null) {
        try {
            mOutputStream.write(buffer);
        } catch (IOException e) {
            Log.e(TAG, "write failed", e);
        }
    }
```

Android - install & debug



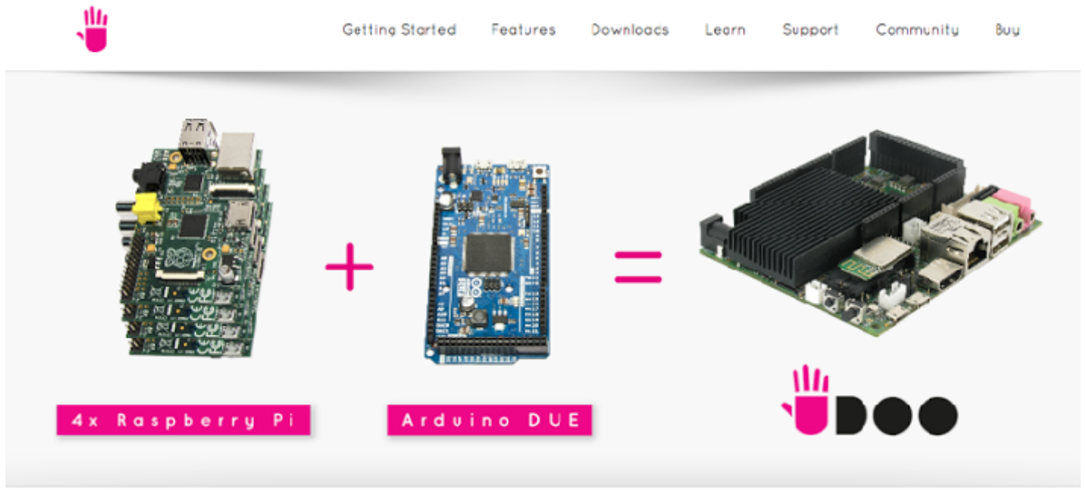
- https://www.youtube.com/watch?v=h3YJjE29YoI



Community

UDOO Community: www.udoo.org

Arduino Community, Rasperry Pi Community and UDOO Community:





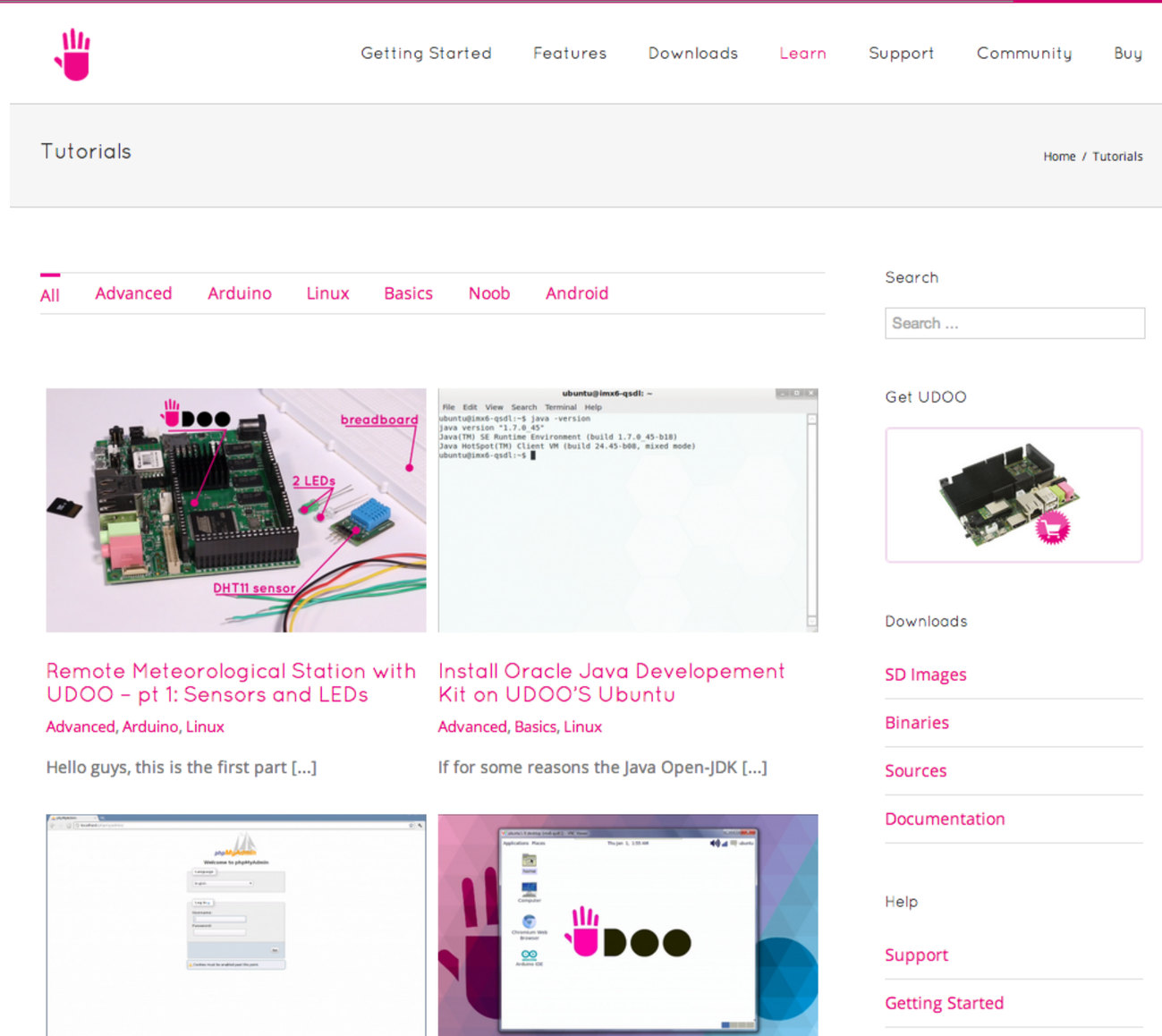


- Informative website
- Forum
- Tutorials
- Projects
- Wiki
- Support
- Faq

www.udoo.org

UDOO Video Tutorials/Projects

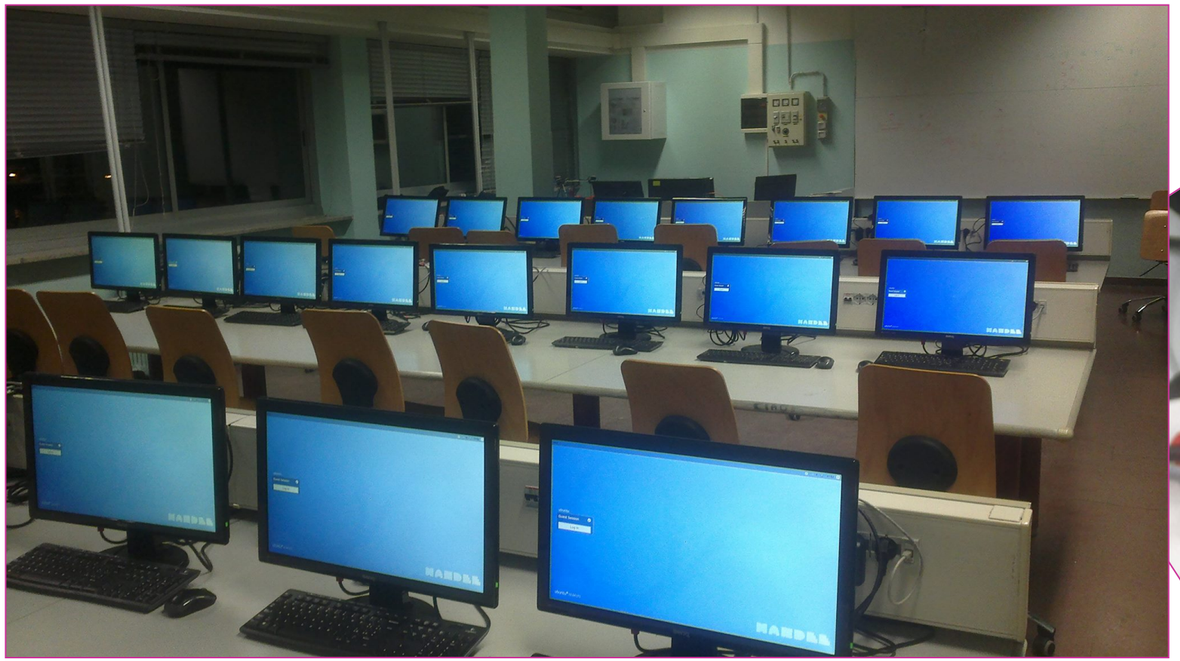
- NAS with OpenMediaVault and “Bazinga!”
- Remote Meteorological Station
- Web Server
- Getting started with Android ADK
- Controlling sensors with Java, Python, …
- …



Beta Tester, Universities and Schools

- Carnegie Mellon - Pittsburgh - USA
- SUPSI - FabLab Lugano - SWITZERLAND
- San diego University - USA
- Instituto Ensenyança Secundaria, Barcelona - SPAIN
- University of Vermont - USA
- University of Siena - ITALY
- Università di Pisa - ITALY
- Ocad university - CANADA
- Aarhus University - DENMARK
- University of Singapore – SINGAPPORE
- University of Florence – ITALY
- Politecnico di Torino – ITALY
- Università di Roma la Sapienza – ITALY
- ESIEE Paris – FRANCE
- Scuola Superiore Sant'Anna Pisa – ITALY
- Politecnico di Milano – ITALY
- I.S.I.S. "Arturo Malignani" di Udine – ITALY
- University of California Berkeley - USA

Computer Lab based on UDOO





Computer Lab based on 30 UDOO at the ISIS Arturo Malignani (Udine, Italy)

www.udoo.org

Community Projects: UDOO Lunar Rover Development

The **Google Lunar XPrize (GLXP)** is offering a Grand Prize of $20 million for safely landing a privately-funded spacecraft on the Moon, driving 500 meters, and sending HD video back to Earth by the end of 2015.

Carnegie Mellon University is developing a lunar rover in partnership with Astrobotic Technology, Inc. to compete for this prize.

The current rover program uses UDOO boards for developing technologies critical to winning the GLXP.

The UDOO was chosen as a development tool for several reasons. First, it has **two processors**, so it can handle processing critical commands and camera data independently. It is also **lightweight (113g)**, **low-power (~2.2W)**, and **user friendly**, enabling students to design and build a fully functioning rover prototype in 90 days. These features are difficult to find in other commercially available computer boards (**15 others were evaluated**).



www.udoo.org

Community Projects: Gapps on UDOO running Android

Thanks to our community member gryphius_g, today we are showing how to quickly install the Gapps in your UDOO, check out the Forum Thread.

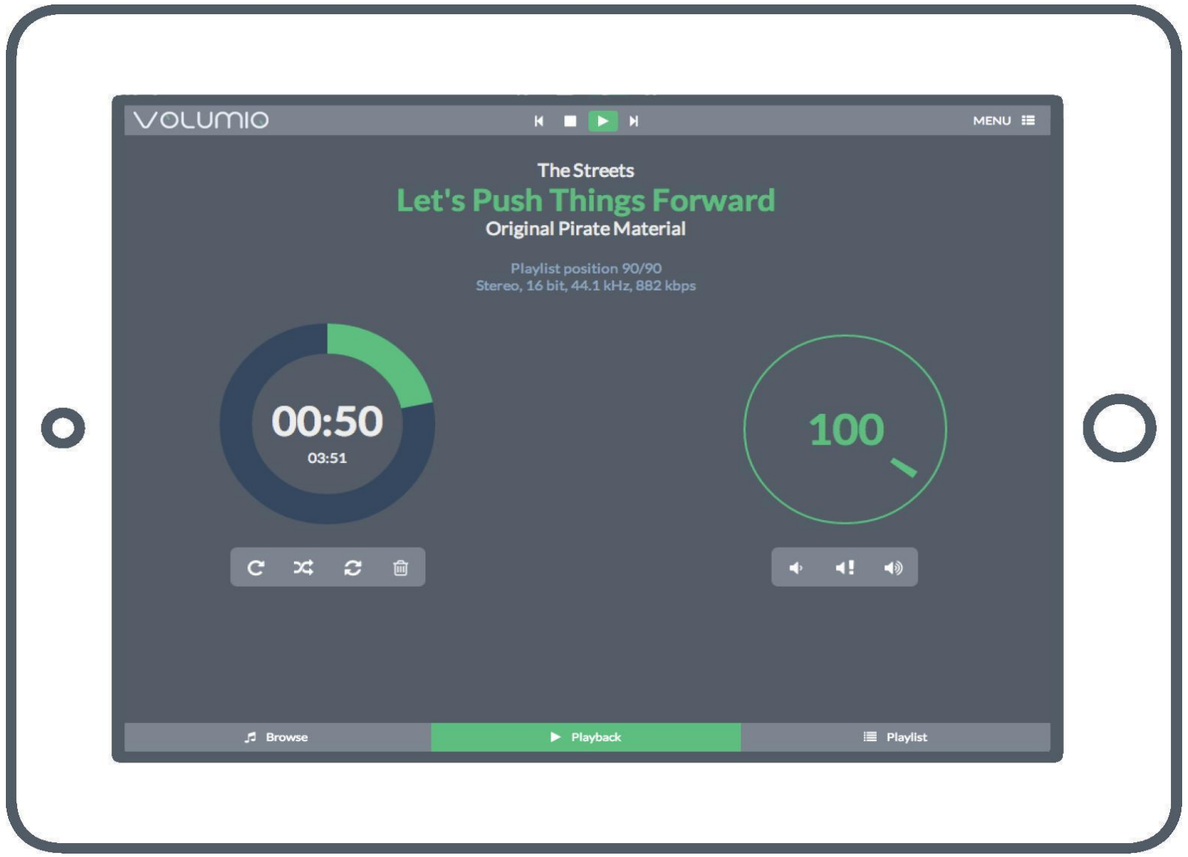


How to Install Gapps on UDOO running Android

www.udoo.org

Community Projects: Volumio, Audiophile Music Player

Volumio is an Audiophile Music Player, specifically designed to **turn Udoo into an headless music player**. Just flash Volumio on SD and in ten minutes you'll be listening to your favourite tunes, with Audiophile Quality!. Volumio is designed to be controlled via your Smartphone, Tablet or PC thanks to its handy WebUi.



www.udoo.org

Community Projects: Pure Data for UDOO

Thanks to **Miller Puckette**, the author of both **Max MSP and Pure Data**, we've got the official version of PD compiled for UDOO.

"Pure Data is an open source visual programming language that allows to create software graphically, without writing lines of code. Pd is used to process and generate sound, video, 2D/3D graphics, and interface sensors, input devices, and MIDI." More info here.



Pure Data for UDOO!

You can download the software here (Stable release, Pd version 0.45-3 compiled for UDOO).

Community Projects: XBMC for UDOO by Stephan Rafin

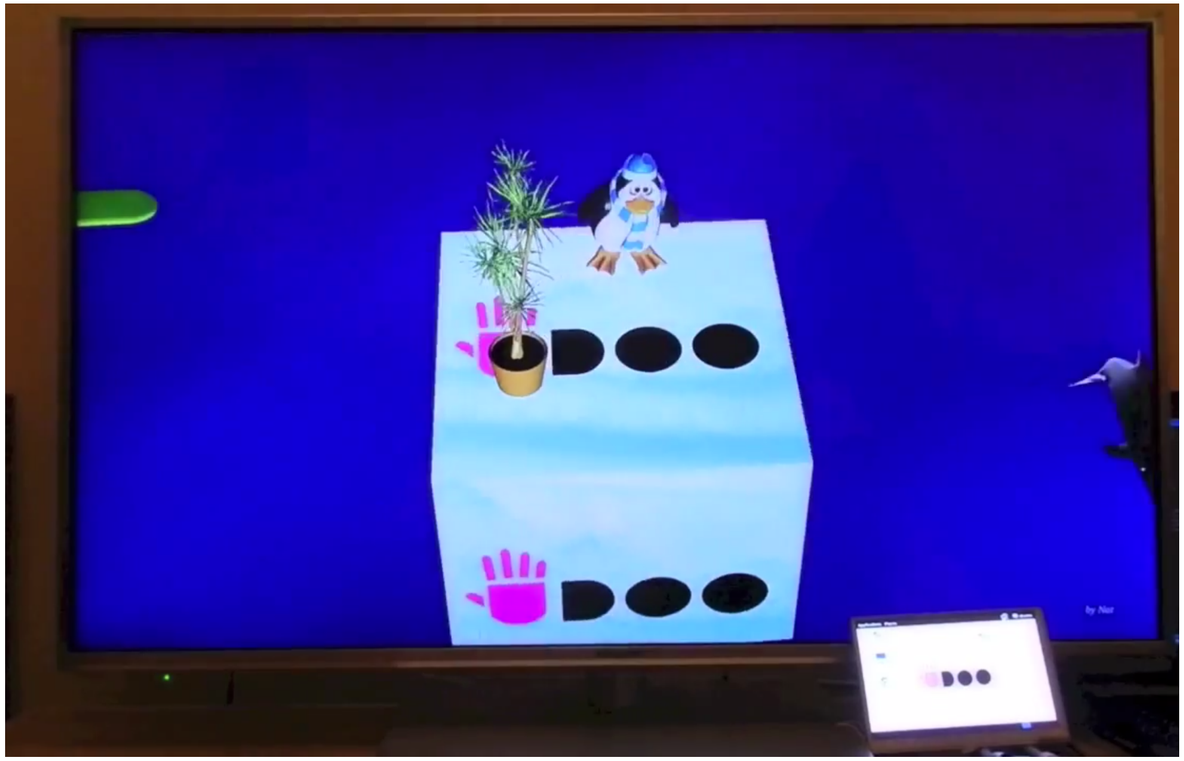
Official XBMC release by Stephan Rafin with full graphic acceleration based on openGL ES 2.0 and can run HD videos very smoothly.



Here for the official thread where you'll find the link to download the image for both the Quad and Dual version: http://Here for the official thread where you'll find the link to download the image for both the Quad and Dual version: http://www.udoo.orgHere for the official thread where you'll find the link to download the image for both the Quad and Dual version: http://www.udoo.

Community Projects: QT 3D with OpenGL ES 2.0

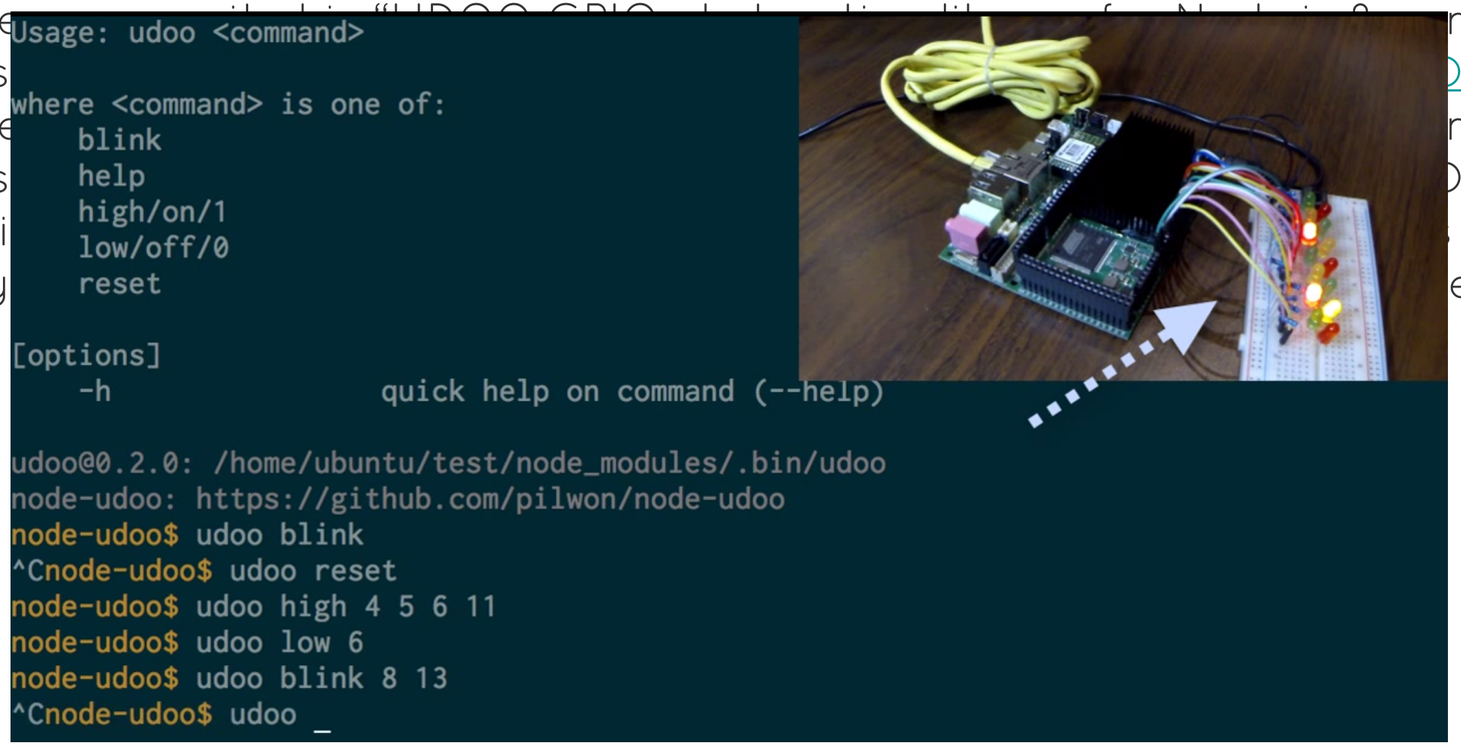
We have compiled Qt with OPENGL ES 2.0 support for UDOO board and we made a funny 3D animation with Qt 3D 2.0 (Qt Quick 2.0 library) to test it. We found it runs very fluently on UDOO i. MX6 Quad board.



https://www.youtube.com/watch?v=AjuPrOd4mtU

Community Projects: Node.js + UDOO by phuh

Our User phuh shared with the community his “UDOO GPIO abstraction library for Our User phuh shared with the community his “UDOO GPIO abstraction library for Node.jsOur User phuh shared with the community his “UDOO GPIO abstraction library for Node.js & command line toolOur User phuh shared with the mmand line tool“. The library has OOO Our User phuh shared with the mmand line tool“. The library has OOO PinoutOur User phuh shared wi & command line tool“. The library e UDOO Pinout Diagram.



Community Projects: Freedomotic Open Source Building Automation

Freedomotic is an open source, flexible, scalable and smashups oriented software that can interact with well known standard building automation protocols as well as with “do it yourself” solutions.

It treats the web, social networks and frontends just like any other sensor or actuator in your automation system. For example you can tweet your washing machine to start the washing cycle. It allows you to build smart spaces. Freedomotic can manage from small apartments to huge buildings, like museums, schools, corporate offices, malls and university campuses.



www.udoo.org