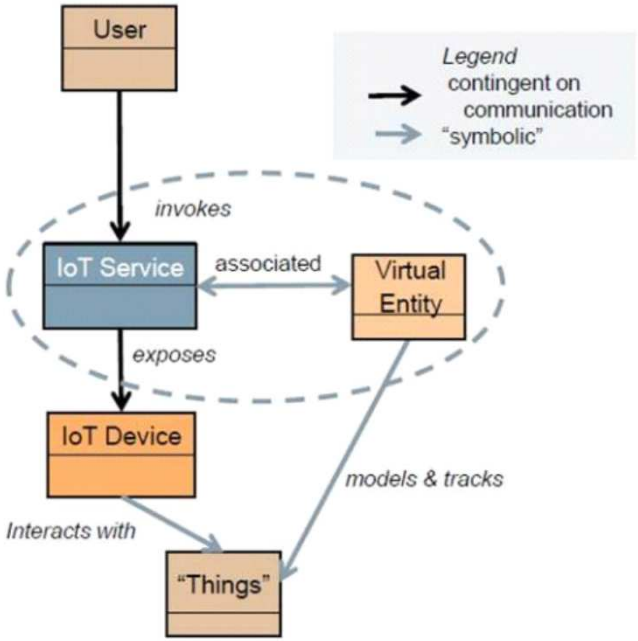
EIOT

IOT Platforms

Aleksander Pruszkowski

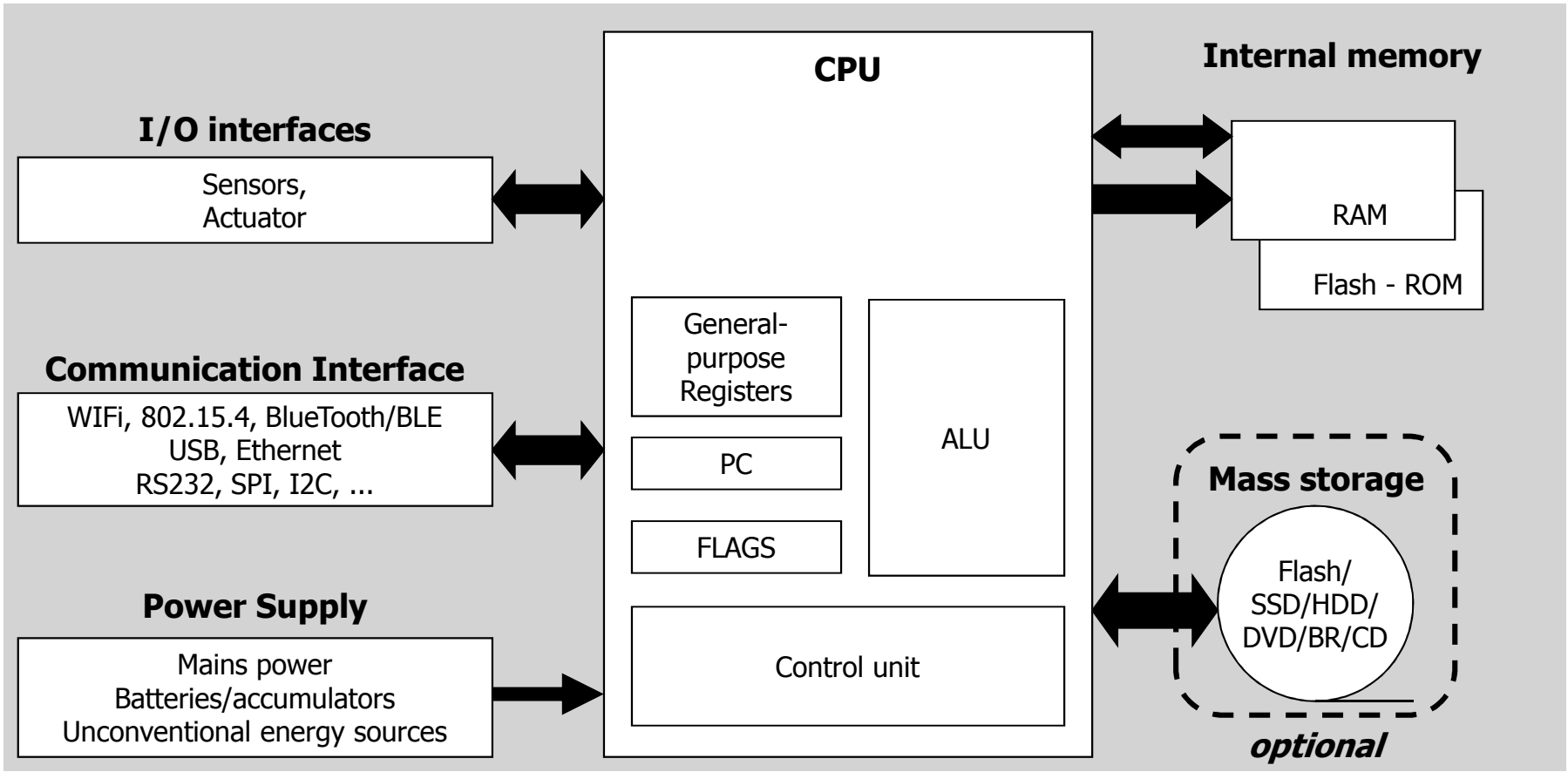
Instytut Telekomunikacji Politechniki Warszawskiej



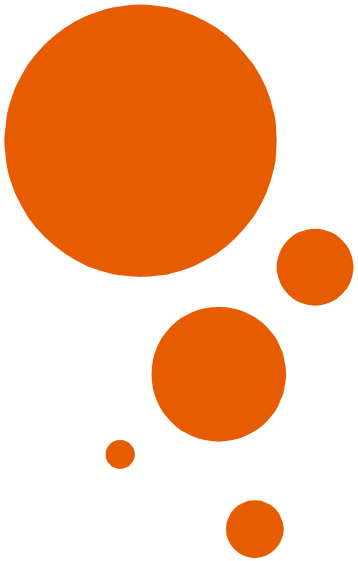
This lecture

- IOT nodes
- Minimal platforms
- Arduino-compliant platforms
- Mid-size platforms
- Linux-based platforms

IOT nodes


CPU
Internal memory
I/O interfaces
Sensors,
Actuator
RAM
Flash - ROM
General-
purpose
Registers
ALU
Communication Interface
WIFi, 802.15.4, BlueTooth/BLE
USB, Ethernet
RS232, SPI, I2C, ...
PC
Mass storage
FLAGS
Flash/
SSD/HDD/
DVD/BR/CD
Power Supply
Mains power
Batteries/accumulators
Unconventional energy sources
Control unit
optional

Minimal platforms

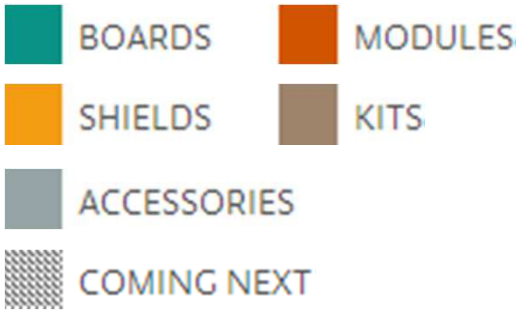


Minimal platforms: Arduino

- Some basic info
  - An outgrowth of students work
  - Arduino is …
    - hardware
      - MCU boards and extension boards (so called shields)
      - standard board pinout (i.e., signals and connectors available for adding components)
      - communication and programming via built-in interfaces (USB, RS232/UART, BT, ...)
      - powered via USB or a power adapter
    - software tools (IDE)
    - collection of libraries

Minimal platforms: Arduino

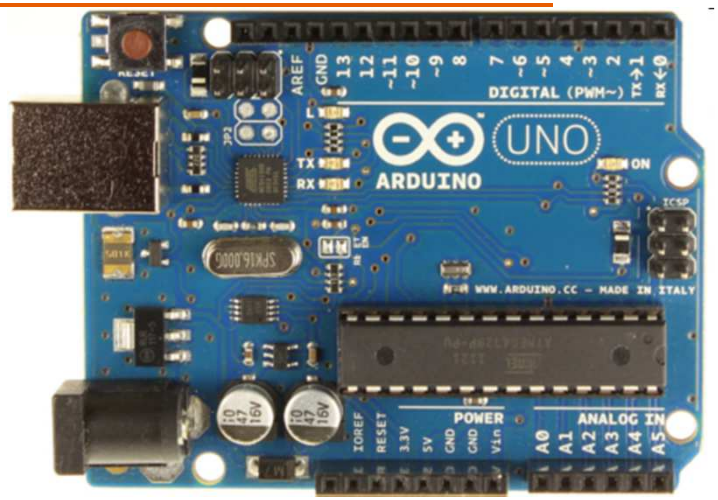
- Hardware: available MCU boards and shields



Źródło: arduino.cc

Minimal platforms: Arduino

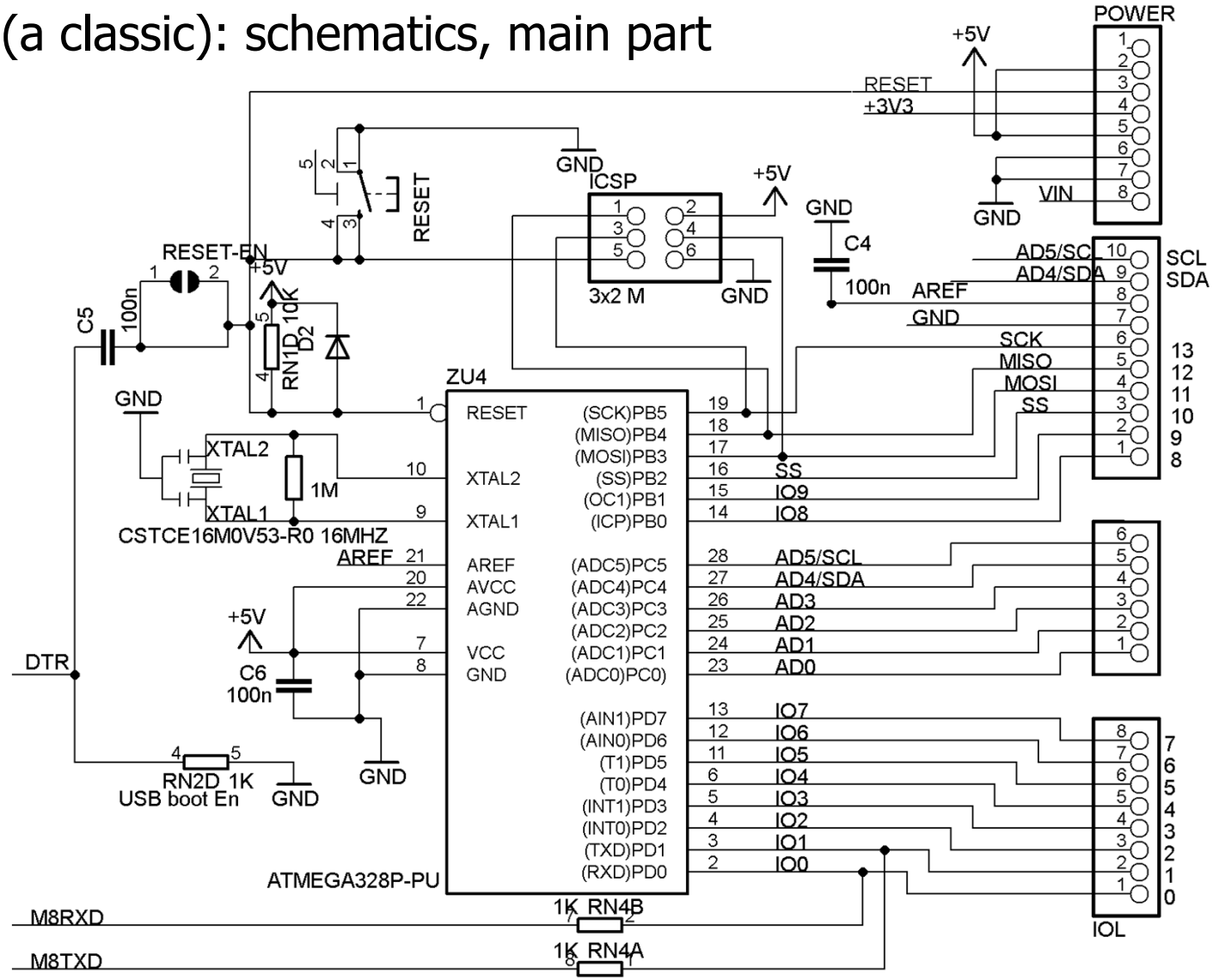
- Arduino UNO (a classic): resources
  - Main CPU: ATmega328P
    - 32KB code memory (incl. 0.5KB bootloader)
    - 2KB data memory
    - 1KB of EEPROM memory
    - clock 16MHz
  - 6 analog inputs
  - 14 digital I/O pins
    - maximum load current per I/O pin 40mA
    - 6 of these supports PWM, working with frequency of 490Hz, 980Hz (pins 5 and 6)
  - peripherals: 1xUART, 1xSPI, 1xI2C, 2x external interrupts
  - CPU-controlled LED diode (LED13)
  - power supply voltage : 7-12V or supply power from PC via USB
  - USB connectivity: only to supply power and emulate a serial port (emulation implemented with another CPU), two diodes ”show” the data flow via the emulated serial port
  - two ISP connectors to program CPUs, one per CPU



Źródło: arduino.cc

Minimal platforms: Arduino

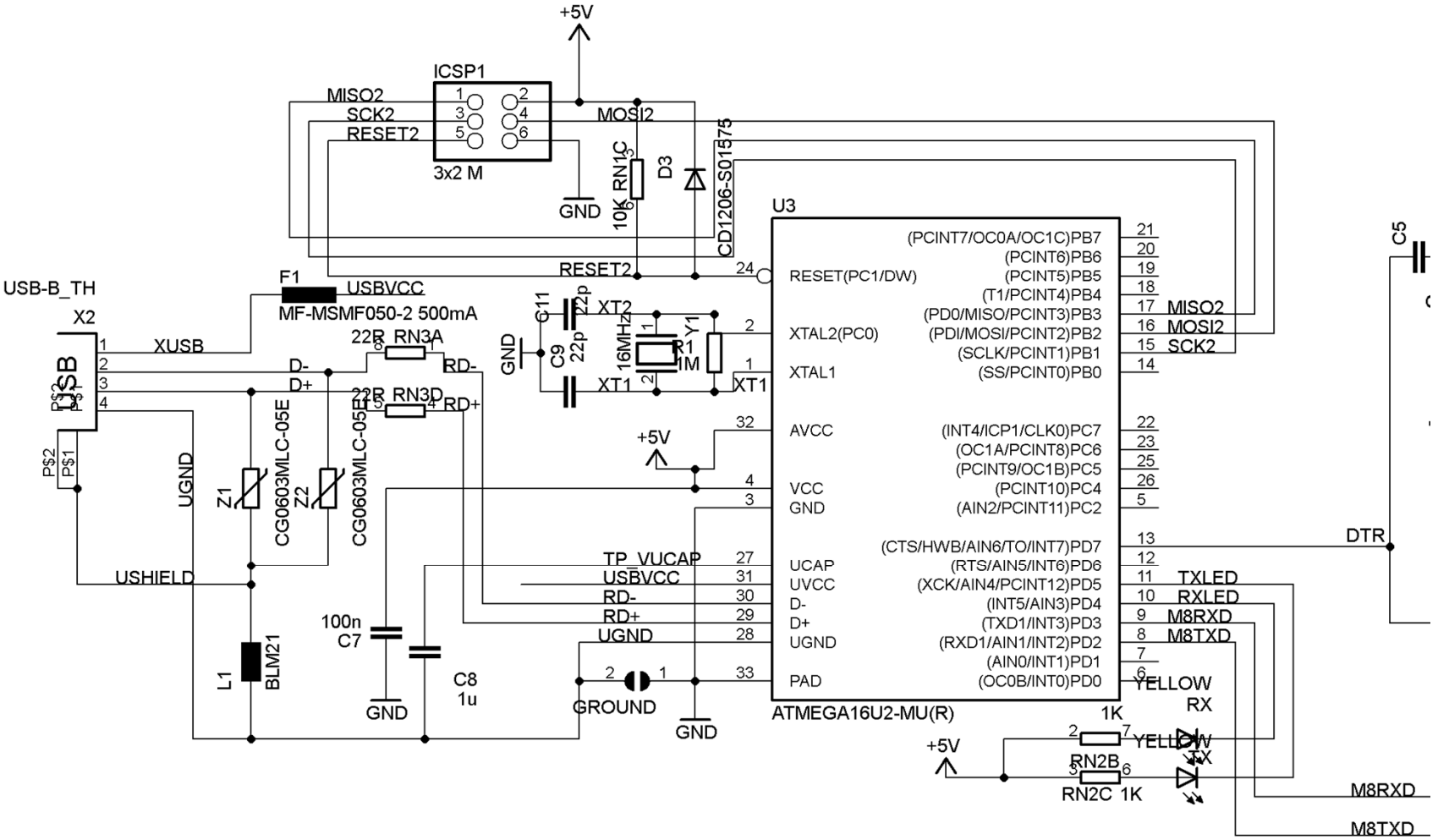
- Arduino UNO (a classic): schematics, main part



Źródło: arduino.cc

Minimal platforms: Arduino

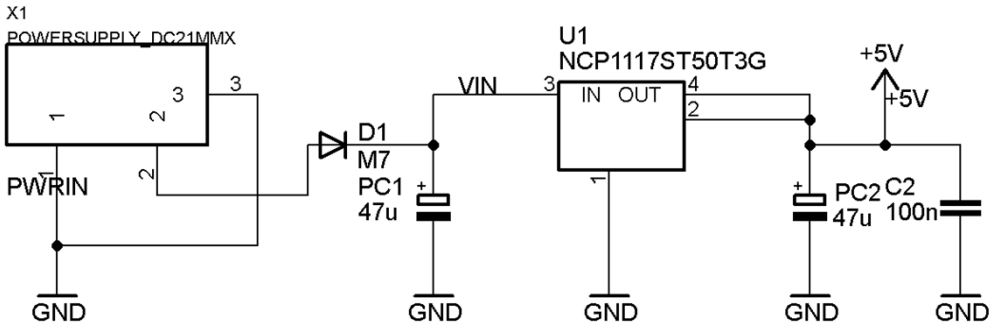
- Arduino UNO (a classic): schematics, usb–uart converter



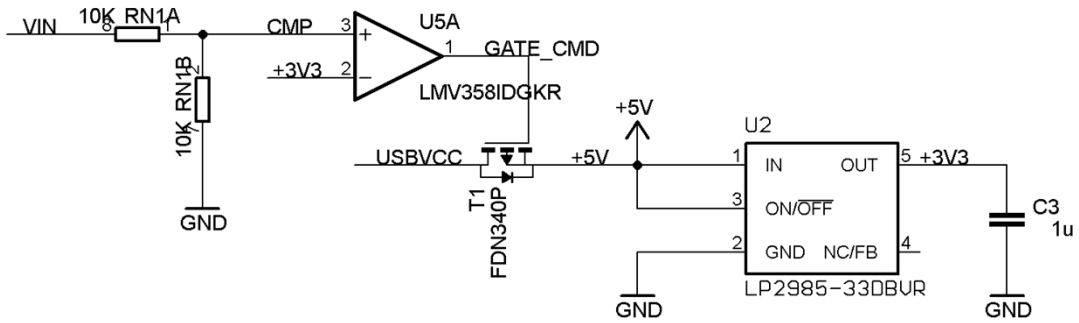
Źródło: arduino.cc

Minimal platforms: Arduino

- Arduino UNO (a classic): schematics, power supply circuitry
  - Generating voltage from an external source



  - Detection of the source of power supply (USB or external) and voltage generation

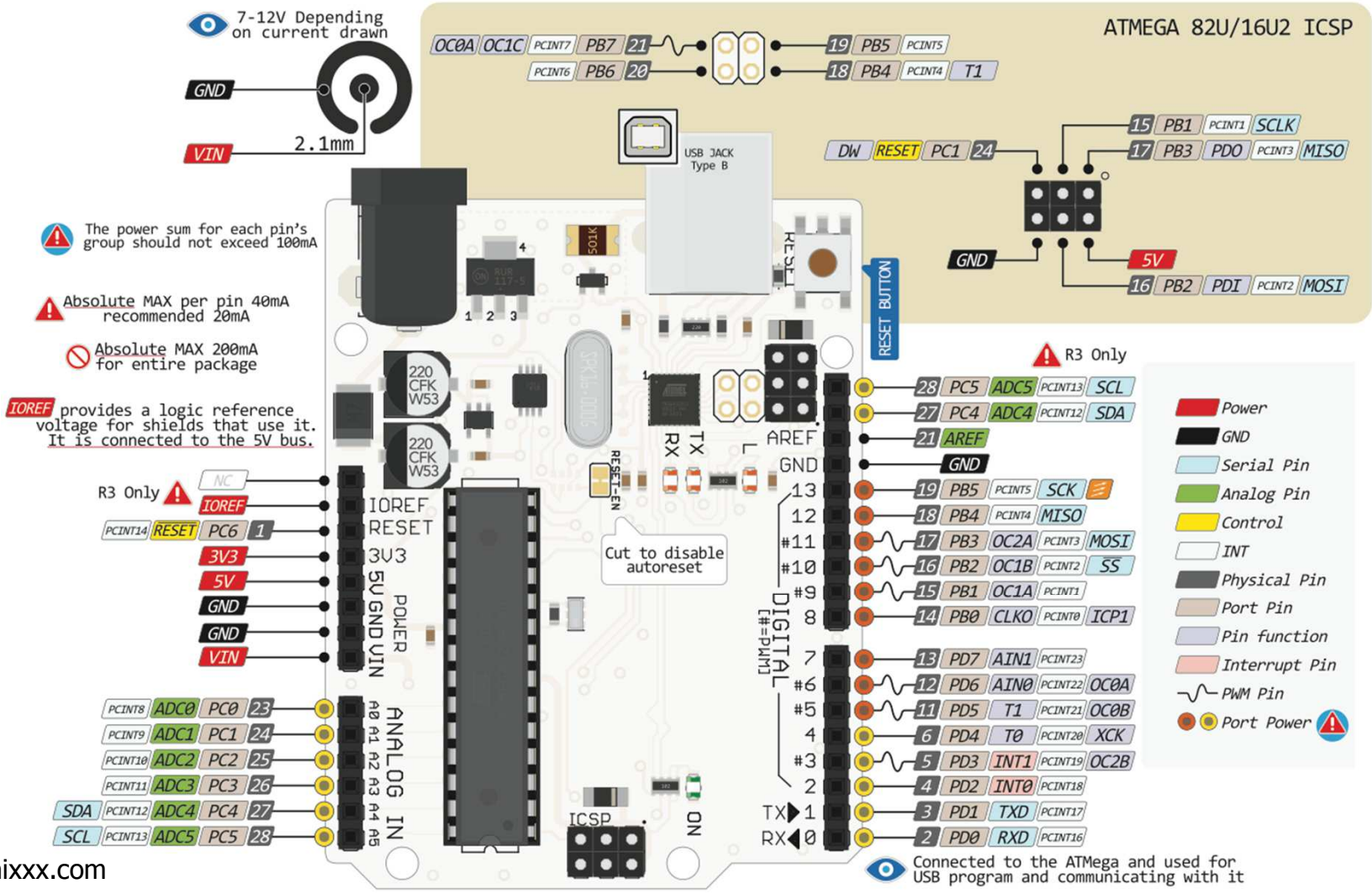


    - Note: Low energy efficiency (NCP1117 introduces voltage drop <1.2V at 800mA and voltage drop on D1 is 0.6V)

Źródło: arduino.cc

Minimal platforms: Arduino

- Arduino UNO (a classic): board pinout



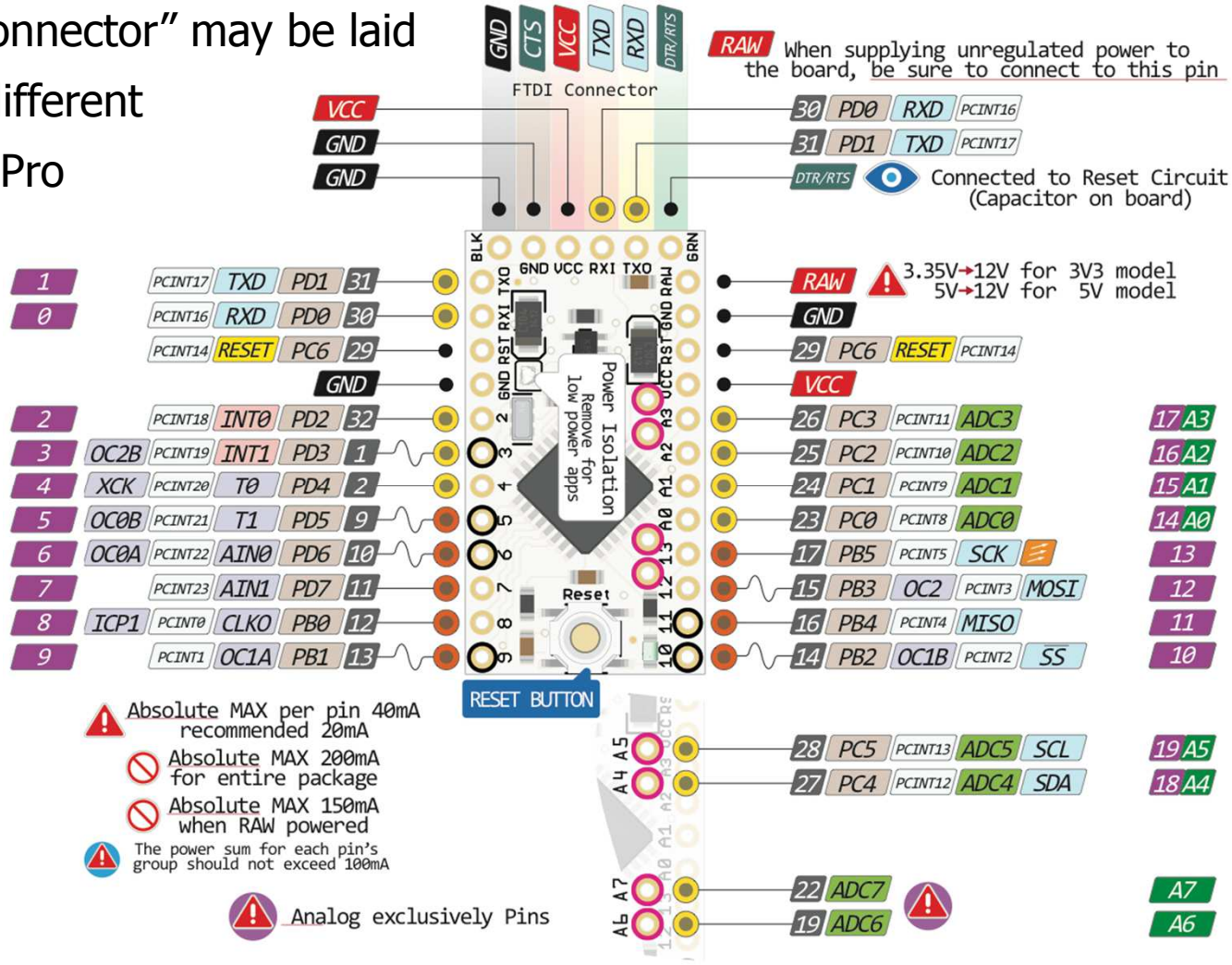
Źródło: www.pighixxx.com

Minimal platforms: Arduino

- Arduino Mini Pro (really small): resources
  - Almost like Arduino UNO
  - Difference with respect to Arduino UNO
    - no USB connector
    - no dedicated connector for power supply (only voltage and ground pins - similar to GPIO)
    - power supply voltage from 3.35 to 12V (3.3V version) or from 5 to 12V (5V version)
    - built-in voltage regulator MIC5205
      - this is a „Low Noise Low Drop" regulator, almost no impact on ADC built-in the CPU
        - for Vref=5V and ADC working in 10bit mode and minimal quantization level is 4mV, then the noise generated by the power supply can be not greater than 400uV
      - voltage regulator current <150mA
        - MCU consumes <12mA (at full speed) ; the regulator offers a big margin
      - voltage drop <350mV
        - CPU can work with power supply voltage as low as 2V; hence Arduino Mini Pro can be powered with voltage as low as 2.4V!
  - clock 8 MHz (3.3V version) or 16 MHz (5V version) – note a bootloader problem!

Minimal platforms: Arduino

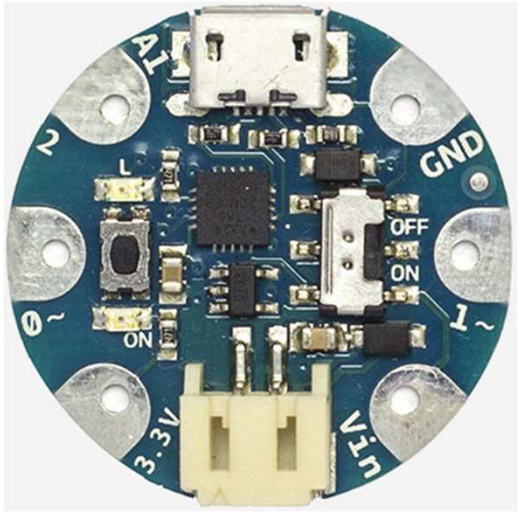
- Arduino Mini Pro (really small): board pinout
  - Pins in „FTDI Connector” may be laid out differently different batches of Mini Pro

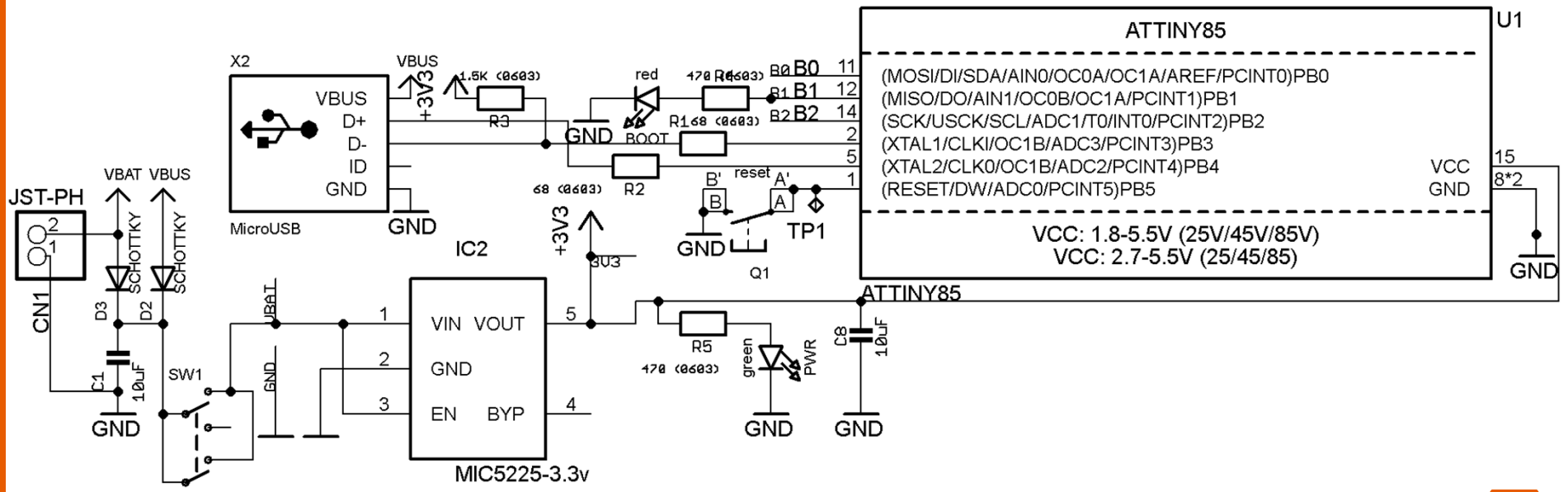


Źródło: www.pighixxx.com

Minimal platforms: Arduino

- Arduino Gemma (wearable module)
  - CPU: ATtiny85 (AVR-compatible)
    - 8KB code memory (incl. 2,75KB for bootloader)
    - 512B data memory, 512B EEPROM memory
    - clock 8MHz, power supply voltage 3.3V
    - 3 digital I/O pins, 2 PWM pins, 1 analog input
  - Current consumption 9mA, diameter ~28mm

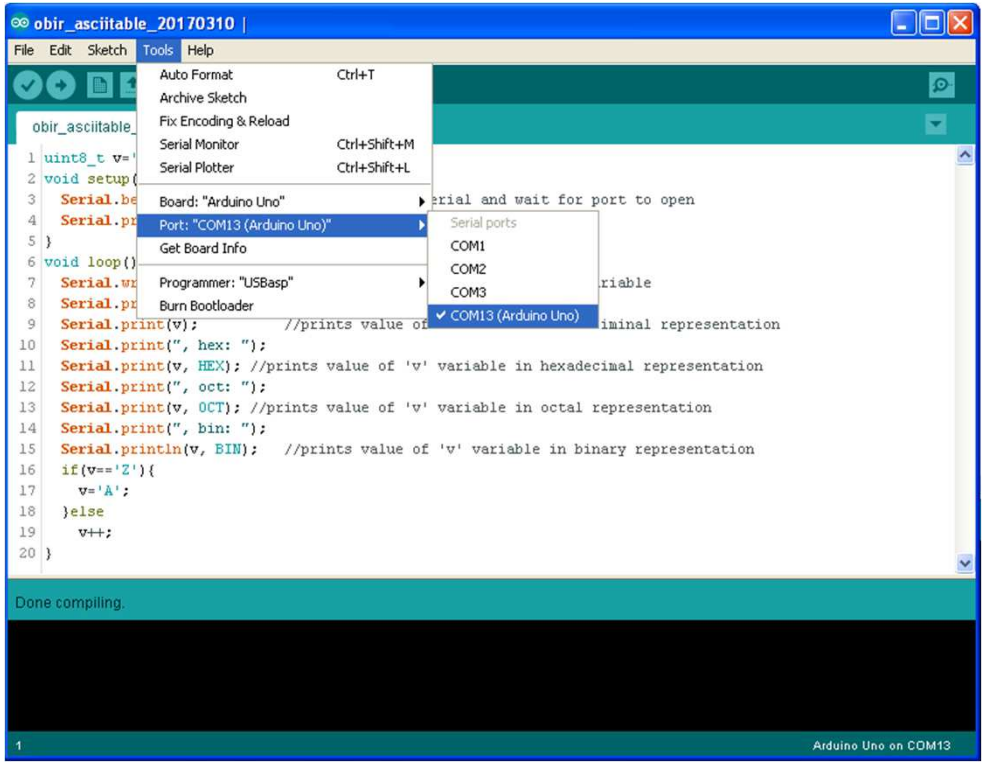




Źródło: arduino.cc

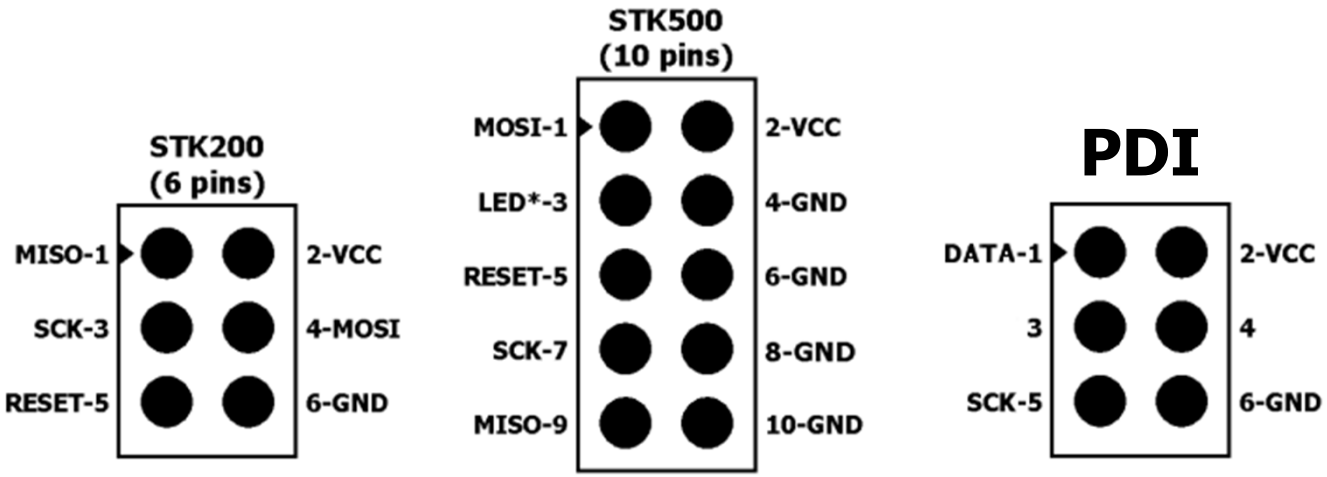
Minimal platforms: Arduino, tools

- Arduino tools
  - dedicated IDE
    - built—in editor
    - integrated preprocessor and compiler
    - tools to program MCUs via ISP
    - serial monitor
  - library collection
    - a wide variety of C++ libraries
      - a new approach to generate code for MCUs!
      - object-oriented programming – peripheral resources modeled as objects
  - Competition:
    - Eclipse C/C++
    - MDK-ARM (Keil)
    - Visual Studio



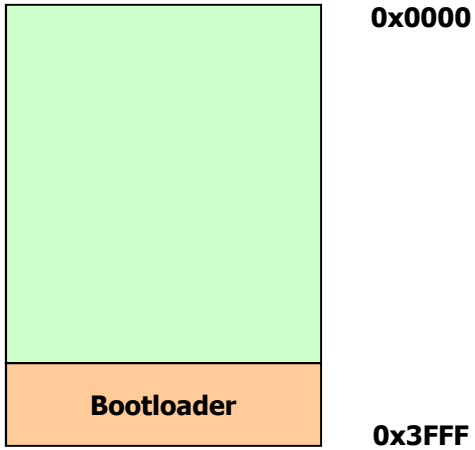
Minimal platforms: Arduino, tools

- Replacing MCU code in AVR-compliant platforms
  - Basic way: ISP - In System Programming
    - no need to remove integrated circuits from the board
    - an alternative way – parallel programmers – are expensive and inconvenient
  - ISP cons
    - one needs a programmer designed specifically for AVR
    - at board design time, one needs to add an additional, special-purpose connector
      - there are many such connectors (e.g., STK200, STK500, PDI, TPI – Attiny)
      - some board designers use their own, proprietary solutions



Minimal platforms: Arduino, tools

- Replacing MCU code in AVR-compliant platforms (cont.)
  - A different way: programming via a communication interface
    - interfaces: serial port, USB and others (say, an Ethernet module)
      - no need to a cable dedicated to programming



Minimal platforms: Arduino, tools

- Replacing MCU code in AVR-compliant platforms (cont.)
  - done with a bootloader installed on the MCU
    - what does the bootloader do? It gets executed from the FLASH memory and allows one to store executable code in the remaining part of the FLASH memory
      - the BOOTRST FUSE bit determines whether, after RESET, CPU starts executing code from the address 0x0000 or from the bootloader's address
      - the size of the bootloader code is "declared" with two FUSE bits: BOOTSZ1, BOOTSZ0
    - Arduino uses the Optiboot bootloader (takes about ~500B of code memory)
    - the serial port DTR signal resets the CPU, which starts the bootloader code
      - the programming commands comply with the STK500 protocol (used in Atmel programmer's)

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	256 words	4	0x0000 - 0x3EFF	0x3F00 - 0x3FFF	0x3EFF	0x3F00
1	0	512 words	8	0x0000 - 0x3DFF	0x3E00 - 0x3FFF	0x3DFF	0x3E00
0	1	1024 words	16	0x0000 - 0x3BFF	0x3C00 - 0x3FFF	0x3BFF	0x3C00
0	0	2048 words	32	0x0000 - 0x37FF	0x3800 - 0x3FFF	0x37FF	0x3800

Źródło: atmel.com

Minimal platforms: Arduino, tools

- Replacing MCU code in AVR-compliant platforms (cont.)
  - tools for code replacement
    - AVRDUDE – a tool used to program many different CPUs
      - supports assorted interfaces: DAPA, STK200, STK500, Altera ByteBlaster, Xil, PAVR, JTAG ICE MKII, RS232, Arduino, USB bootloader (based on V-USB), ...
      - integrated with Arduino IDE
      - the Arduino bootloader supports only a fraction of AVRDUDE capabilities(!)
    - other tools: UISP, PonyProg, AtmelStudio (an IDE combo), and many others

Minimal platforms: Arduino, programming

- Arduino applications
  - the sketch

```
void setup() {
  //initializing code executed only once
}
void loop() {
  //code executed repeatedly
}
```

what does it really look like?

```
...
int main(void){
  init();
  ...
  setup();
  ...
  for (;;){
    ...
    loop();
  }
  ...
  return 0;
}
```

  - example: simple reading from a temperature sensor (serial communications)

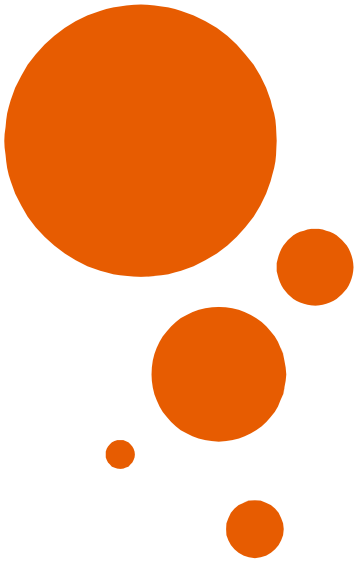
```
#include <NewSoftSerial.h>
NewSoftSerial therm(6,5);      //"serial" thermometer connected to pins 6 i 5
...
void loop() {
  therm.listen();
  if(therm.available() > 0){
    int c = therm.read();
     ...
  }
}
```

  - for more examples, see introductions to the EIOT labs or visit arduino.cc

Minimal platforms: Arduino, programming

- Advantages of using the Arduino ”ecosystem”
  - the market-driven concept of de facto Arduino compatibility
    - for example, peripherals made as Arduino ”shield” coming Arduino-compatible software drivers (libraries)
  - low entry barriers
    - free and ready-to-use tools (C/C++ compiler, programmers)
    - a board includes both the CPU (MCU) and a means to program it
    - cheapest boards start at about $10 (as of May 2018)
  - lots of hardware and software on the market
    - multiple Arduino clones
    - a great collection of libraries to use in your projects
      - however, be wary of errors in libraries and possible problems with stable coexistence of different libraries

Mid-size Platforms

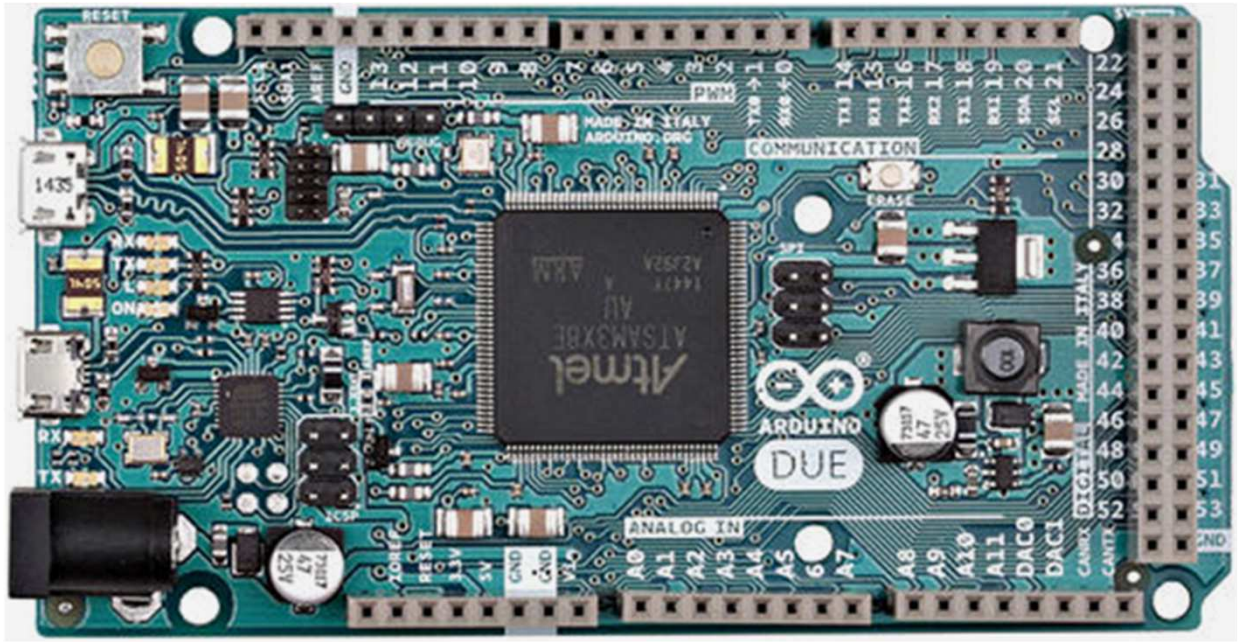


Mid-size platforms

- Mid-size refers to both a platform's processing power and its memory
  - say, ROM and RAM capacity both less than 20MB
  - these platforms cannot run „typical" operating systems (e.g., Linux), due to too limited resources, especially RAM memory
- The most meaningful classification of these platforms is likely based on their CPU:
  - ARM-based
  - based on other CPU architectures (Intel, ESP8266, MIPS, ...)

Mid-size platforms - ARM

- Products by Arduino: Arduino Due
  - CPU: AT91SAM3X8E (32bits, ARM Cortex-M3)
    - 512KB code memory, 96KB data memory in two separate blocks: 64KB and 32KB)
    - clock: 84MHz
  - 54 digital I/O pins; 12 of those support PWM
  - 12 analog inputs
  - 4xUART(3.3V), DMA, 2xI2C, SPI, CAN, 2xUSB (device-USB and USB-OTG)
  - Programming via ATmega16U2 (USB-UART converter)
  - Connectors work at 3.3V voltage technology



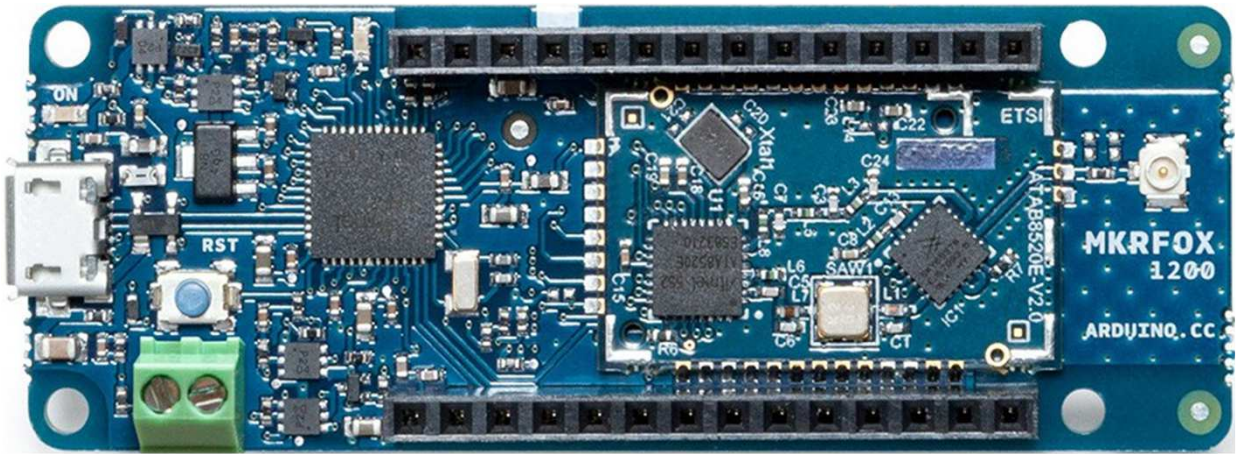
Źródło: arduino.cc

Mid-size platforms - ARM

- Products by Arduino: Arduino MKRFOX 1200
  - CPU: ATSAMD21G18 (32bity, ARM Cortex M0+)
    - 256KB code memory
    - 32KB data memory
    - clock: 48MHz, RTC: 32768Hz
  - Battery-powered: 2xAA(AAA)
  - 1xUART, 1xI2C, 1xUSB (programming and communicating with a PC)
  - Built-in interface supporting the SigFox radio networks
    - currently, the SigFox radio module communicates with CPU via UART and AT commands
  - Twin board: Arduino

MKRFOX WAN 1300

has LoraWAN support

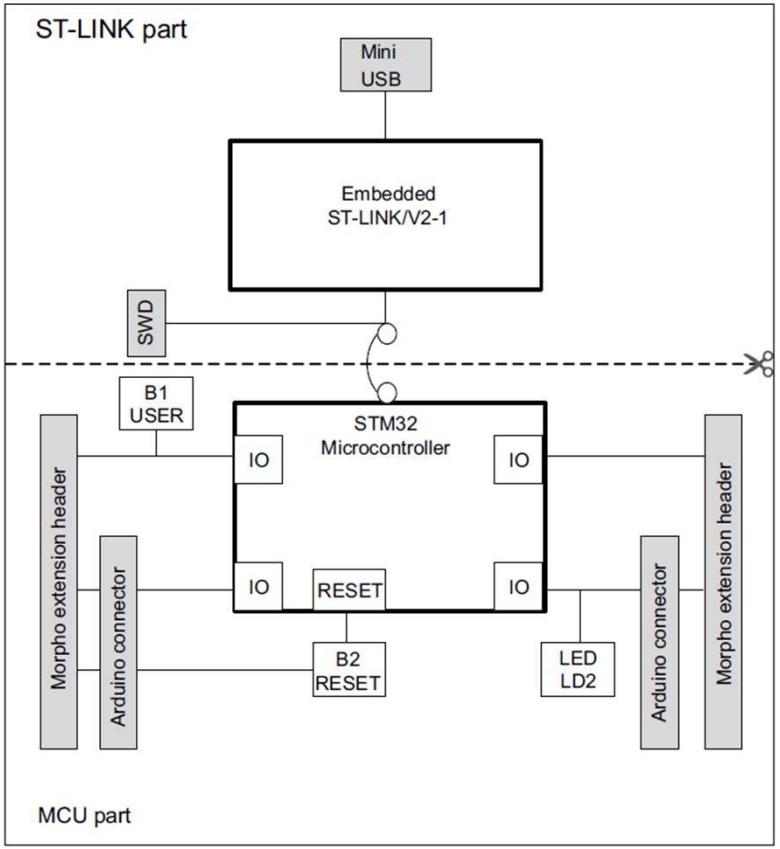


Źródło: arduino.cc

Mid-size platforms - ARM

- STM32 Nucleo
  - extends the Arduino pinout standard by adding extra Morpho pins
  - compatible with an Internet-based ARM code development platform:

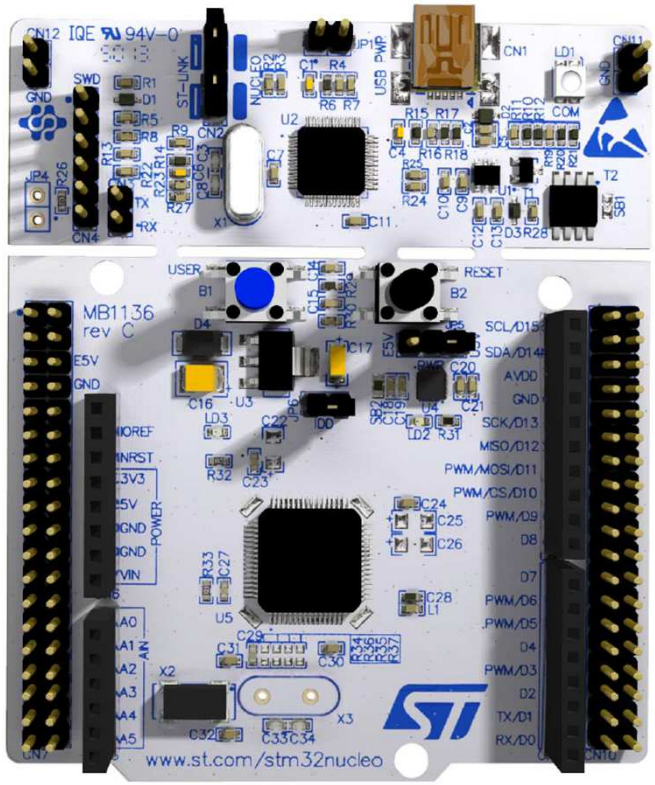
    MBED.org
  - built-in SWD programmer and debugger
  - the maker provides HAL libraries



Źródła: st.com

Mid-size platforms - ARM

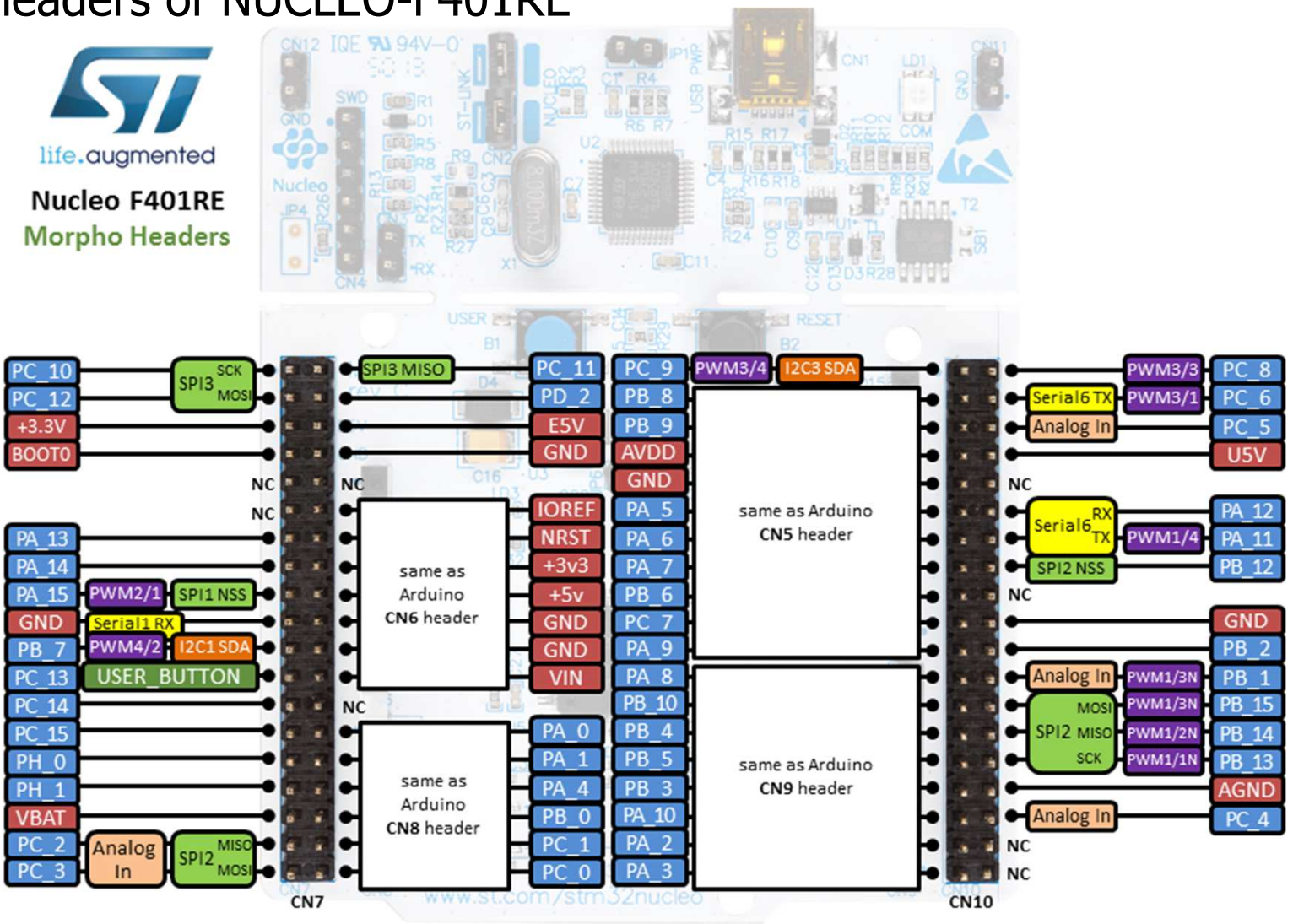
- NUCLEO-F401RE
  - CPU: STM32F401RET6 (32bity, ARM Cortex-M4 + FPU)
    - 512KB FLASH (code) memory
    - 96KB data memory
    - clock frequency up to 84MHz, RTC
    - 50 digital I/O pins (+ interrupt generation)
    - 16 analog inputs (12 bit)
    - 3xSPI , 3xI2C , 4xUSART , 1xSDIO (SD card interface), 1xUSB/OTG/FS
    - 7 general-purpose counters
  - USB
    - programming (built-in SWD programmer: ST-Link)



Źródła: st.com

Mid-size platforms - ARM

- Morpho headers of NUCLEO-F401RE



Źródła: st.com

Mid-size platforms - ARM, tools

- Replacing MCU code
  - via the JTAG interface (Joint Test Action Group)
    - a serial interface (signals: TCK, TMS, TDI, TDO, TRST)
      - JTAG-tested integrated circuits can form a chain
      - initially JTAG was used to test connections on PCBs (printed circuit boards)
    - when it comes to CPU, JTAG makes possible the following (and more):
      - starting/stopping/stepping through/ code execution
      - inspecting and initializing content of registers and memory
        - a more exotic operation: initialize the CPU's instruction register, IR , with the opcode of some instruction; the instruction will be executed in the next machine cycle (so called code injection)
    - to use JTAG you need a hardware JTAG programmer
      - some platforms (SAME70, STM32Discovery) are equipped with such a programmer
        - a USB connector, once connected with a PC, serves both as JTAG connector and COM/UART (the latter enables user application-level communication with the platform)

Mid-size platforms - ARM, tools

- Replacing MCU code, cont.
  - via the JTAG interface (Joint Test Action Group), cont.
    - some well-known tools that enable working with JTAG
      - CMSIS-DAP(mbed), CoLinkEX(Coocox), I-jet(IAR), J-Link(Segger), LPC-LINK(NXP), ST-LINK(STMicroelectronics), ULINK(Keil)
      - OpenOCD – a free replacement for commercial products
    - each of the above tools requires configuration files (!)
      - one for each CPU type and for each interface type
      - in OpenOCD the configuration files describe
        - an interface (way how the debugged system is attached)
        - a board
        - a CPU and target memory

Mid-size platforms - ARM, tools

- Replacing MCU code, cont.
  - replacing code via mass storage emulation
    - an MCU or board becomes a (small size) mass storage (as seen by a PC), in which the PC may store new firmware
  - replacing code via a built-in bootloader (residing in MCU's ROM memory)
    - example: a program stored in ROM of the AT91RM9200 (ARM) MCU can load code, retrieving it from one of built-in communication modules (UART, I2C, SPI, USB)

Mid-size platforms: ESP8266-based

- ESP8266-based platforms
  - first modules, ESP-01, made by AI-Thinker, available in 2014
    - their firmware supported AT commands (used in modems), which made it possible to configure WiFi and use a TCP/IP stack
  - MCU (the heart of the module): ESP8266
    - made by Espressif Systems
    - CPU: 32bit RISC Tensilica Xtensa L106, clock 80MHz (up to 160MHz)
    - ROM: 64KB, data: 96KB, can support external memory (up to 16MB) to store the code (through Q-SPI interface)
    - communication: IEEE 802.11 b/g/n (supports WPA and WPA2)
    - 16 digital I/O, SPI , I2C , UART , one 10-bit ADC (0...1V)
  - programming
    - no(!) dedicated operating system
    - NodeMCU (the Lua language), Arduino IDE (ESP8266 Arduino Core), MicroPython, Espruino (emulating Node.js)
    - ESP-Open-SDK, ESP-Open-RTOS (derived from FreeRTOS)

Mid-size platforms: ESP8266-based

- ESP8266-based AI-Thinker platforms


ESP-01


ESP-03


ESP-05


ESP-07


ESP-08

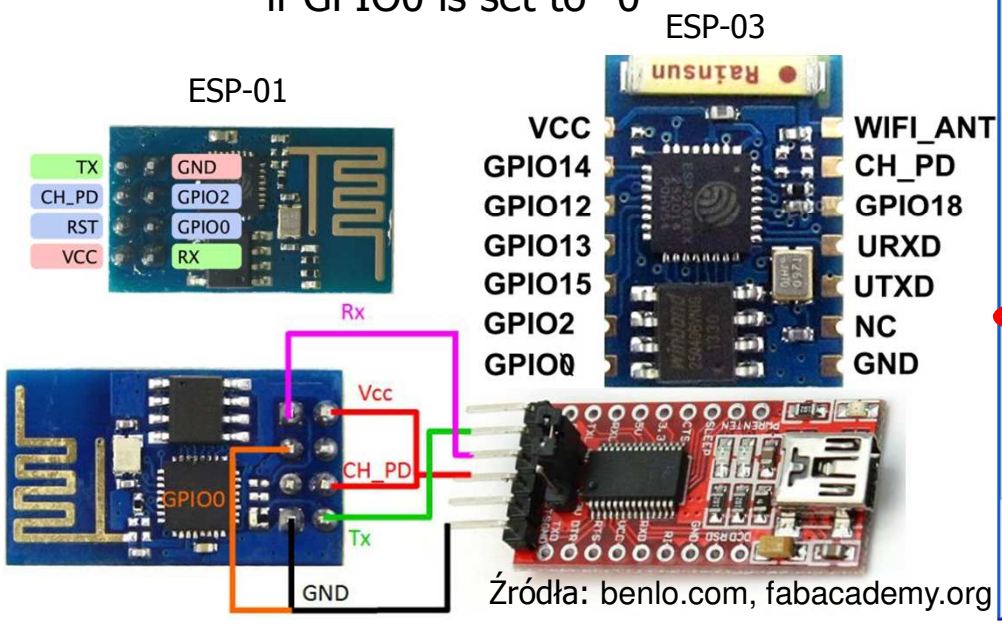

ESP-09

Name	Active pins	Pitch	Form factor	LEDs	Antenna	Shielded?	Dimensions (mm)	Notes
ESP-01	6	0.1 in	2×4 DIL	Yes	PCB trace	No	14.3 × 24.8	
ESP-03	10	2 mm	2×7 castellated	No	Ceramic	No	17.3 × 12.1	
ESP-05	3	0.1 in	1×5 SIL	No	U.FL connector	No	14.2 × 14.2	
ESP-07	14	2 mm	2×8 pinhole	Yes	Ceramic + U.FL connector	Yes	20.0 × 16.0	Not FCC approved
ESP-08	10	2 mm	2×7 castellated	No	None	Yes	17.0 × 16.0	Not FCC approved
ESP-09	10	misc	4×3 dice	No	None	No	10.0 × 10.0	
ESP-12	14	2 mm	2×8 castellated	Yes	PCB trace	Yes	24.0 × 16.0	FCC and CE approved[14]
ESP-12E	20	2 mm	2×8 castellated	Yes	PCB trace	Yes	24.0 × 16.0	4 MiB Flash
ESP-13	16	1.5 mm	2×9 castellated	No	PCB trace	Yes	W18.0 × L20.0	Marked as "FCC". Shielded module is placed sideways, as compared to the ESP-12 modules.
ESP-14	22	2 mm	2×8 castellated +6	No	PCB trace	Yes	24.3 × 16.2	

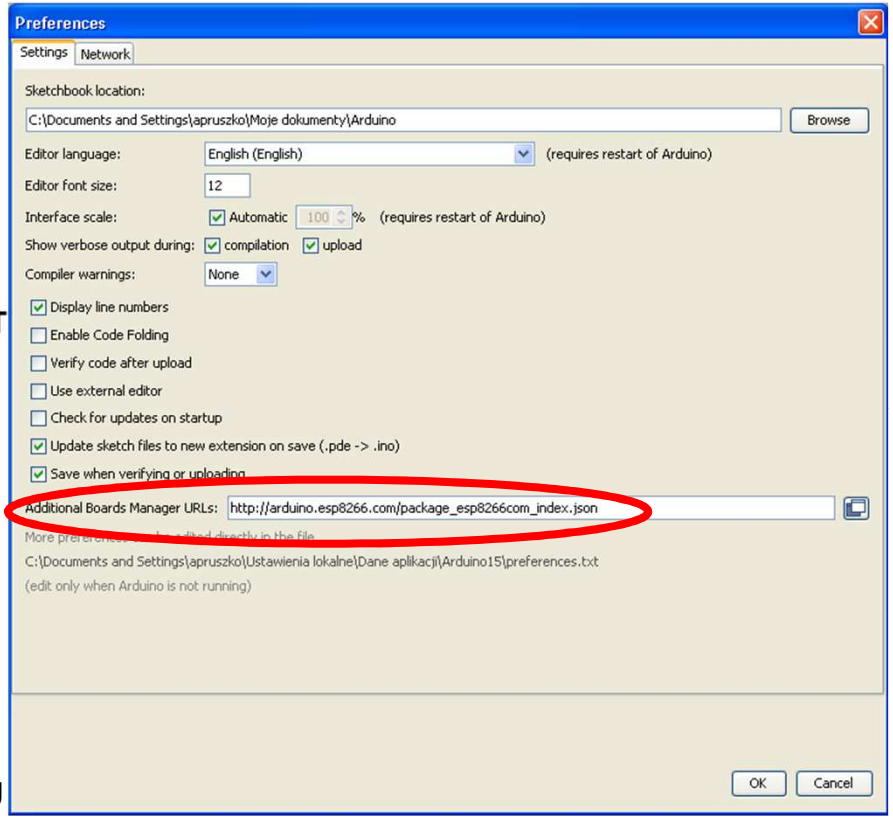
Źródła: wikipedia.org, adrianandgenese.com

Mid-size platforms: ESP8266-based

- ESP8266-based AI-Thinker platforms, cont.
  - to use Arduino IDE, one needs to provide the Board Manager a new URL, e.g.,
    http://arduino.esp8266.com/stable/package_esp8266com_index.json
  - for the module to work, one needs to supply power; then setting the CH_PD pin to ”1” turns the module on
    - setting the pin to ”0” makes the module enter a sleep mode
  - code is replaced via the RX i TX pins, if GPIO0 is set to ”0”

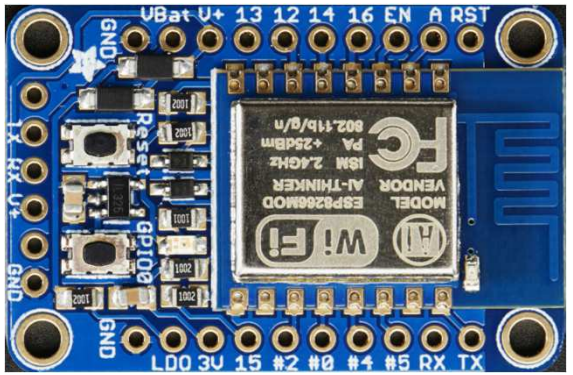


Źródła: benlo.com, fabacademy.org



Mid-size platforms: ESP8266-based

- Some well-known ESP8266-based platforms
  - Adafruit HUZZAH ESP8266
    - you can insert the module in a breadboard
    - additional features:
      - RESET and USER buttons (ease code replacement)
      - voltage level shifter (from 5V to 3.3V, because of UART)
      - 3.3V voltage regulator for supplying power



  - a list of other ESP8266-based platforms is quite long:
    - Arduino Uno WiFi (a hybrid of Atmega328 & ESP8266), SparkFun ESP8266 Thing, ESP8266 NodeMcu LUA V3, ESP8266 Witty Mini NodeMCU, WiFi Bee ESP8266, WeMos D1 R1 WiFi ESP8266, D1 mini WiFi ESP8266 IoT, Wio Node WiFi ESP8266 IoT, ...
- Problems with ESP8266
  - SDK used by Arduino IDE is not fully open; some libraries are precompiled and distributed only as binaries
  - the module consumes a lot of energy (up to 250mA at 3.3V)
  - when you use it or develop your own code for it, you may encounter unstable behavior, with unknown causes

Źródło: adafruit.com

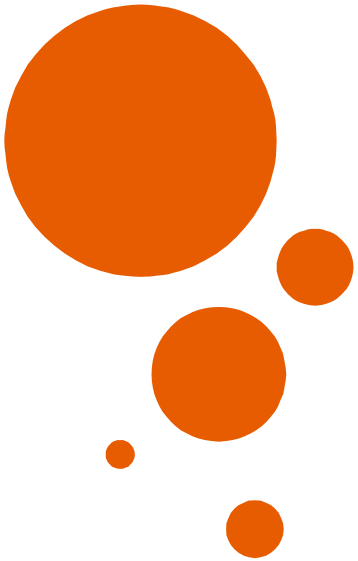
Mid-size platforms: ESP8266-based, code

- Programming a ESP8266-based platform with Arduino IDE
  - example: an MQTT client (taken from Arduino IDE examples)

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

#define MY_LATENCY  10000
#define MSG_LEN     25

#define MQTT_SUB    "esp8266/sub„
#define MQTT_PUB    "esp8266/pub"

char cID[]="ESP8266_0000";

WiFiClient espClient;

PubSubClient client(espClient);

long timeofLastMsg = 0;
char msg[MSG_LEN];

void my_reconnect() {
  while (!client.connected()) {
    if (client.connect(cID))
      client.subscribe(MQTT_SUB);
    else
      delay(5000);
  }
}
```

```
void my_callback(char *t, byte *p, unsigned int l){
  //...
}
void setup() {
  WiFi.begin("myssid", "myssid_pass");
  while (WiFi.status() != WL_CONNECTED)
    delay(500);
  client.setServer("192.168.1.1", 1883);
  client.setCallback(my_callback);
  pinMode(A0, INPUT);
}
void loop() {
  if (!client.connected())
    my_reconnect();

  client.loop();
  if(timeofLastMsg<millis()){
    snprintf(msg, MSG_LEN,"ADC:%d", analogRead(A0));
    client.publish(MQTT_PUB, msg);
    timeofLastMsg=millis()+MY_LATENCY;
  }
}
```

Linux-based platforms



Linux-based platforms

- Linux-based platforms
  - A wide selection available: Raspbery PI, Intel Edison, Intel Galileo, BeagleBone, Orange PI, NanoPI, Orange PI, Artietta, UDOO, ...
  - Hybrids – combining Linux with "bare bone" MCU (like ATtmega328p)
    - Arduino Yún/Mini/Industrial
    - Arduino Tian
  - In many IoT setups, one uses a home router as the hardware and OpenWRT as software
    - Linksys/Cisco np.: WRT54GL (CPU: MIPS BCM47xx , RAM: 16MB RAM, FLASH: 4MB)
    - TPLink np.: TL-MR3020 (CPU: Atheros AR9331 400MHz, RAM: 32MB, FLASH: 4MB)
    - ...
    - Dedicated embedded platforms based on OpenWRT: Propox MMnet1001 (CPU: AT91SAM9 G20 200MHz, RAM: 64MB RAM, FLASH: 4GB)

Linux-based platforms: Raspberry PI

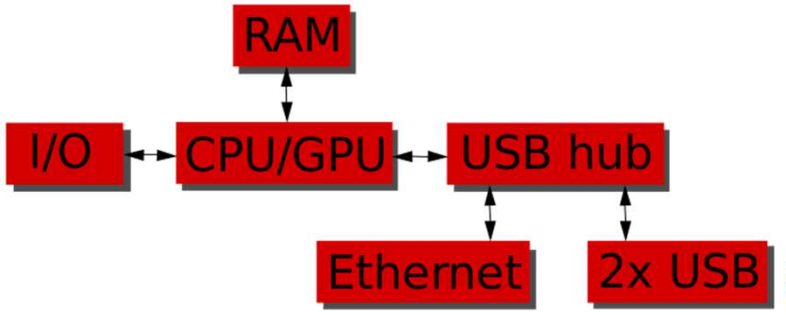
- Raspberry PI
  - a low-cost platform
    - for many years priced at 5$/25$/35$ (2020.05.10)
  - started in 2011
    - the idea dates back to 2006

  - Raspberry Pi is a full-featured computer, supported by a community of enthusiasts
    - you can download images of many operating systems: Debian, Fedora, Ubuntu, Arch, RISC OS, ...
      - installing an image is as simple as transferring it onto an SD card (e.g.,: the ”dd” or ”win32disk” commands)

Źródło: wikipedia.org

Linux-based platforms: Raspberry PI

- Raspberry PI, cont.
  - SoC (**S**ystem **o**n **C**hip) Broadcom; depending on the Raspberry Pi version, one of the following: BCM2835/BCM2836/BCM2837
    - CPU: ARM11/ARM Cortex-A7/ARM Cortex-A53
    - GPU (graphics processing unit: OpenGL, HDMI 1080p, H.264 video decoding)
    - SDRAM controller
    - Host-USB controller
  - SDRAM memory 256MB/512MB/1GB/…/4GB (depending on the version), SD card connector, Ethernet controller (not available in versions A and Zero)
    - code execution starts from the SD card, which simplifies learning the system
      - you cannot damage the device by incorrectly programming non-volatile memory
  - Power supply voltage: 5V, maximum power consumption (depending on the version): 0.8W...12.5W
  - dimensions: 85mm X 53mm (ver. B)
    - version RaPi Zero: 65mm X 30mm



Źródło: wikipedia.org

Linux-based platforms: Raspberry PI

- Raspberry PI, interfacing sensors and actuators
  - P1 connector
    - versions Zero, 2, and 3 have a 40 pin connector
  - how to handle it in software
    - libraries for C, Python, Java, Perl, Ruby
    - example in Java
      - rpi-gpio-java librarry

```
GpioGateway gpio = new GpioGatewayImpl();

//as output pin
gpio.setup(Boardpin.PIN11_GPIO17, Direction.IN);
//as input pin
gpio.setup(Boardpin.PIN12_GPIO18, Direction.OUT);

//read value
boolean input_value=gpio.getValue(
                              Boardpin.PIN11_GPIO17);
//write value (true->"1")
gpio.setValue(Boardpin.PIN12_GPIO18, true);
```

version Pi B

Źródło: wikipedia.org

Linux-based platforms: Raspberry PI

- Raspberry PI, interfacing sensors and actuators, cont.
  - how fast can your drive I/O pins?
  - experimental results*):

Programming language	How obtained	Square waveform
Shell (cmd. tools)	gpio (WiringPi)	40Hz
Shell (”proc FS”)	via access to /proc/...	2,8kHz
Python	wiringpi2	28kHz
Python	RPi.GPIO	70kHz
C	wiringPi	4,1...4,6 MHz
C	Biblioteka BCM2835	5,4MHz
C	Native library**)	14MHz

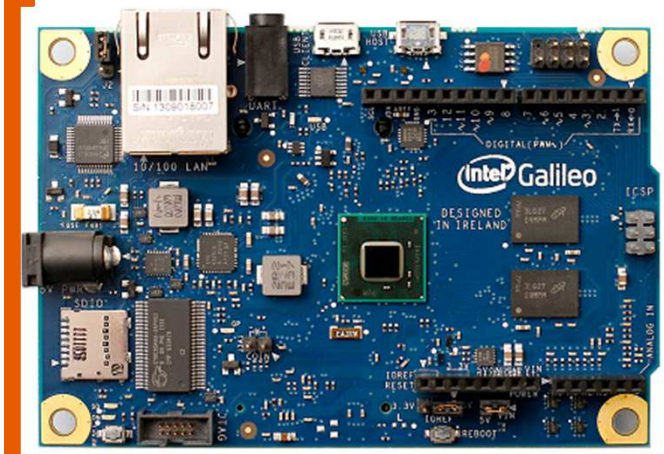
*) Data based on tests: http://codeandlife.com/2012/07/03/benchmarking-raspberry-pi-gpio-speed/

**) Direct access to MCU’s control registers (without any OS overhead, much like in Arduino)

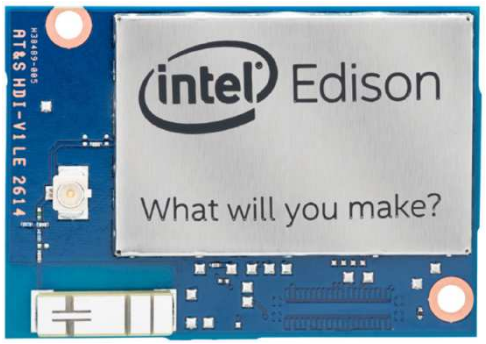
Linux-based platforms: Intel Galileo/Edison

- Intel Galileo gen. 1 (gen. 2) and Edison
  - Edison is a SoC module that requires a ”hosting board”
    - Intel Edison Board for Arduino, Intel Edison Breakout Board
  - Possibility of expansion with peripheral elements
    - 13x digital I/O, 6x analog inputs (12-bit ADC), 6x PWM, 1x SPI, 1x I2C, 1x UART, Ethernet, ...
    - using built-in Host-USB controller, you can connect …
      - WebCam, USB2RS232 converters, Flash memory, 3G/4G memory, ...

Arduino Intel Galileo gen 1

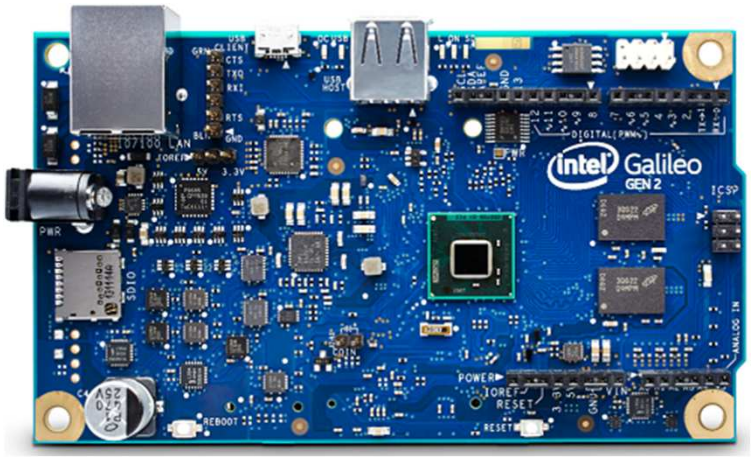


Edison



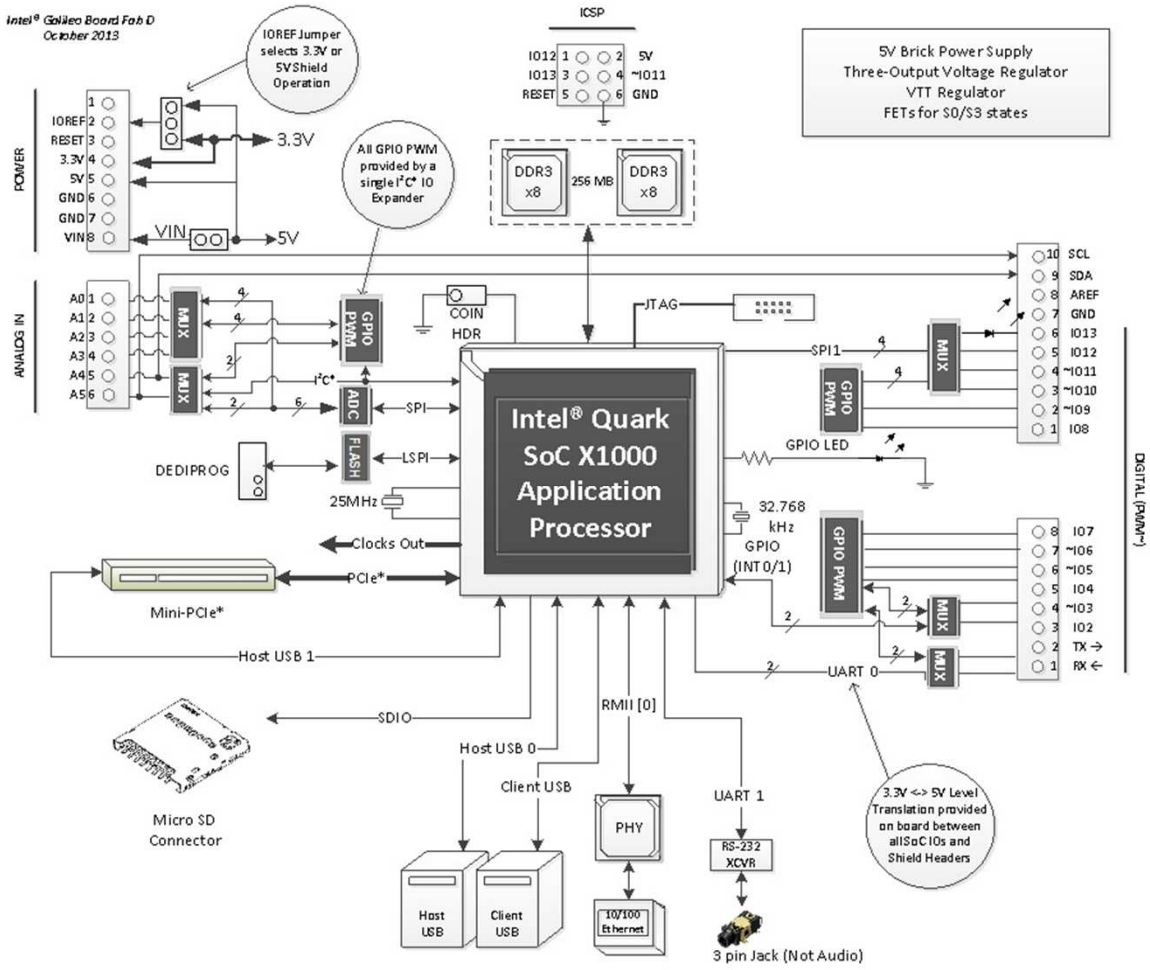
Źródło: communities.intel.com

Arduino Intel Galileo gen 2



Linux-based platforms: Intel Galileo/Edison

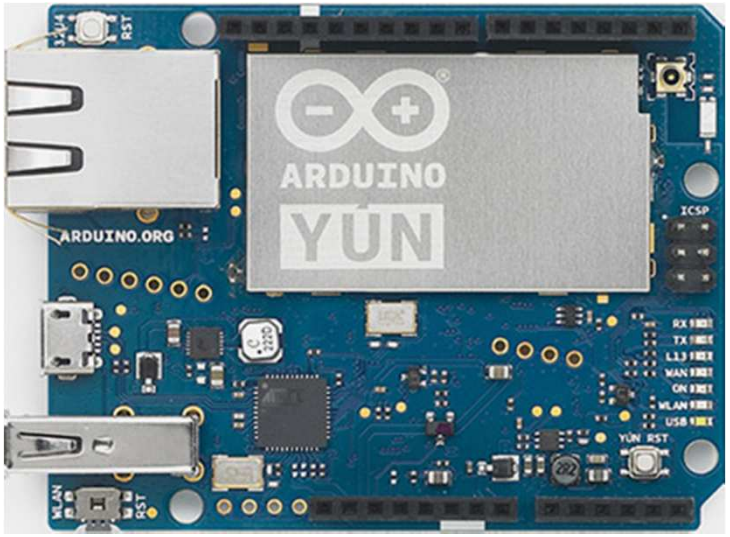
- Intel Galileo/Edison: architecture
  - CPU complies with x86 (32bity)
    - Quark SoC X1000
  - CPU clock: 400MHz
  - memory: 256MB (DRAM)
  - micro-SD connector (up to 32GB) to store the image of an operating system (a native one, Linux, or another one)
  - Galileo/Edison is an "Arduino Certified product”
    - can use Arduino logo and be sold by Arduino Store on the Arduino web site



Źródło: communities.intel.com

Linux-based platforms: Hybrids

- Arduino Yun – two cooperating MCUs on a single board
  - part I (Linux based part)
    - based on Linux (OpenWrt-Yun)
      - can log in with SSH and install new software via 'opkg', ...
    - CPU: Atheros AR9331 (MIPS 24Kc)
    - clock: 400MHz
    - RAM: 64MB
    - FLASH: 16MB
  - part II ("bare bone" MCU)
    - programmed with Arduino IDE
    - CPU: ATmega32U4
    - clock: 16MHz
    - RAM: 2.5KB
    - FLASH: 32KB
  - the cooperation of the two parts is handled by the Bridge library



Źródło: arduino.cc

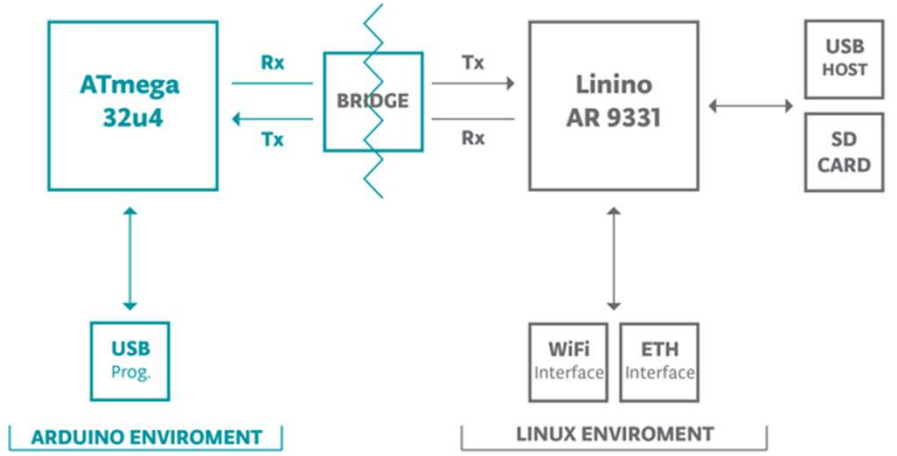
Linux-based platforms: Hybrids

- Arduino Yun: bridging software
  - example of using the Bridge library: getting a page from a WWW server (the Arduino part)

```
#include <Process.h>
void setup() {
  Bridge.begin();
  SerialUSB.begin(9600);
  while (!SerialUSB);
  runCurl();
}
void loop() {//empty!}
void runCurl(){
  Process p;
  p.begin("curl");
  p.addParameter("http://www.arduino.cc/asciilogo.txt");
  p.run();
  while (p.available() > 0)
        SerialUSB.print(p.read());
  SerialUSB.flush();
}
```

initializing the infrastructure connecting the two parts

run Linux command

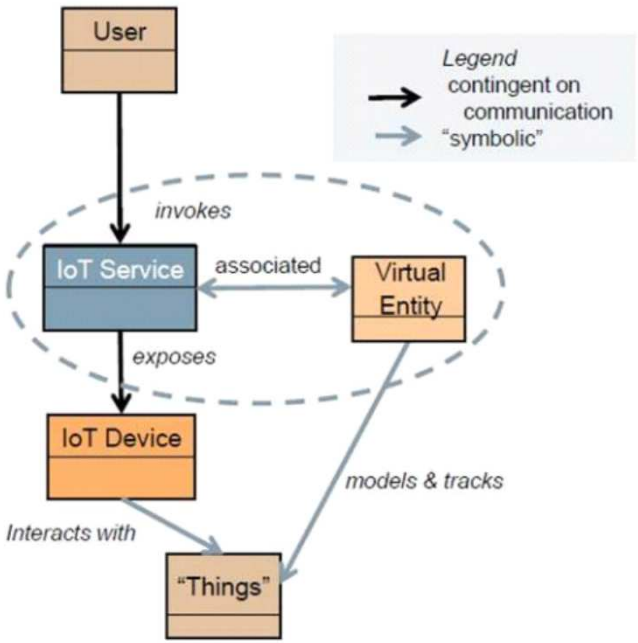


invoking a service

processing the result produced by the service

Źródło: arduino.cc

Thank you!


User
Legend
contingent on
communication
"symbolic"
invokes
IoT Service
associated
Virtual
Entity
exposes
IoT Device
models & tracks
Interacts with
"Things"