

Systemy Wbudowane

dr inż. Adam Mrozek amrozek@agh.edu.pl p. 414/B5 (ETI, ITn)
dr inż. Marek Wilkus (ITn)

Program przedmiotu - wykłady - ETI

- **Wprowadzenie, charakterystyka systemów wbudowanych**
- **Technika cyfrowa**
- **Elementy architektury komputerów**
- **Mikrokontroler ARM Cortex**
 - porty I/O
 - przerwania
 - układ czasowy (timer)
 - interfejsy komunikacyjne
 - obsługa wybranych urządzeń I/O, modulacja szerokości impulsu

Laboratorium 1 – 3 (x86-64) - ETI

- architektura x86-64
- **Linux, GCC (assembler i C/C++)**
- analiza i modyfikacja prostych programów napisanych w assemblerze
 - podstawowe instrukcje i rejestry procesora
 - alokacja i dostęp do pamięci, sterowanie przebiegiem programu
 - stos
- budowa pliku wykonywalnego i bibliotek

Laboratorium 4 – 7 (ARM) - ETI

- mikrokontroler z rdzeniem ARM-Cortex M4 (architektura ARMv7-M)
- środowisko Keil uVision 5 (język C)
- porty I/O
- przerwania
- pomiar czasu
- komunikacja szeregową

Strona przedmiotu oraz regulamin laboratoriów:

galaxy.agh.edu.pl/~amrozek

Zaliczenie: 0 – 10 punktów z każdej z dwóch części.

Oceniana jest każda sekcja w trakcie (pod koniec ćwiczeń).

Liczbę punktów za dane ćwiczenie ustala prowadzący.

Program przedmiotu (wykłady i laboratoria) - ITn

2 wykłady / 2 laboratoria - Adam Mrozek

- **Wprowadzenie, charakterystyka systemów wbudowanych**
- **Podstawowe układy peryferyjne mikrokontrolera (na przykładzie ARM Cortex M4)**
 - porty I/O, przerwania, układ czasowy (timer)

4 wykłady / 4 laboratoria - Marek Wilkus

- **Mikrokontrolery AVR, platforma Arduino**
 - podstawowe peryferia mikrokontrolera, komunikacja szeregową
 - rozbudowa podstawowego układu o dodatkowe elementy: moduły, czujniki itp.

Literatura

Catsoulis J., Designing Embedded Hardware, 2nd Edition, O'Reilly Media, 2008.

Noergaard T., Embedded Systems Architecture 2nd Edition – A Comprehensive Guide for Engineers and Programmers, Elsevier, 2005

Yiu J., The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors 2nd Edition, 2010

Wszelkie książki poświęcone elektronice i technice cyfrowej:

Pieńkos J., Turczyński J., Układy scalone TTL w systemach cyfrowych, WKiŁ

Horowitz P., Hill W., Sztuka Elektroniki, wyd. 12., WKŁ, 2018

Dokumentacja producentów elementów elektronicznych

(+przykłady, aplikacje, w zasadzie wszystko czego potrzeba...)

ARM: www.arm.com developer.arm.com

ST: www.st.com

Atmel: www.atmel.com

www.microchip.com

Podstawowe pojęcia / przypomnienie

Czym jest...

- komputer
- architektura komputera
- procesor
- mikroprocesor
- mikrokontroler
- System on Chip (SoC)

?

Wreszcie:

czym jest system wbudowany?

System Wbudowany

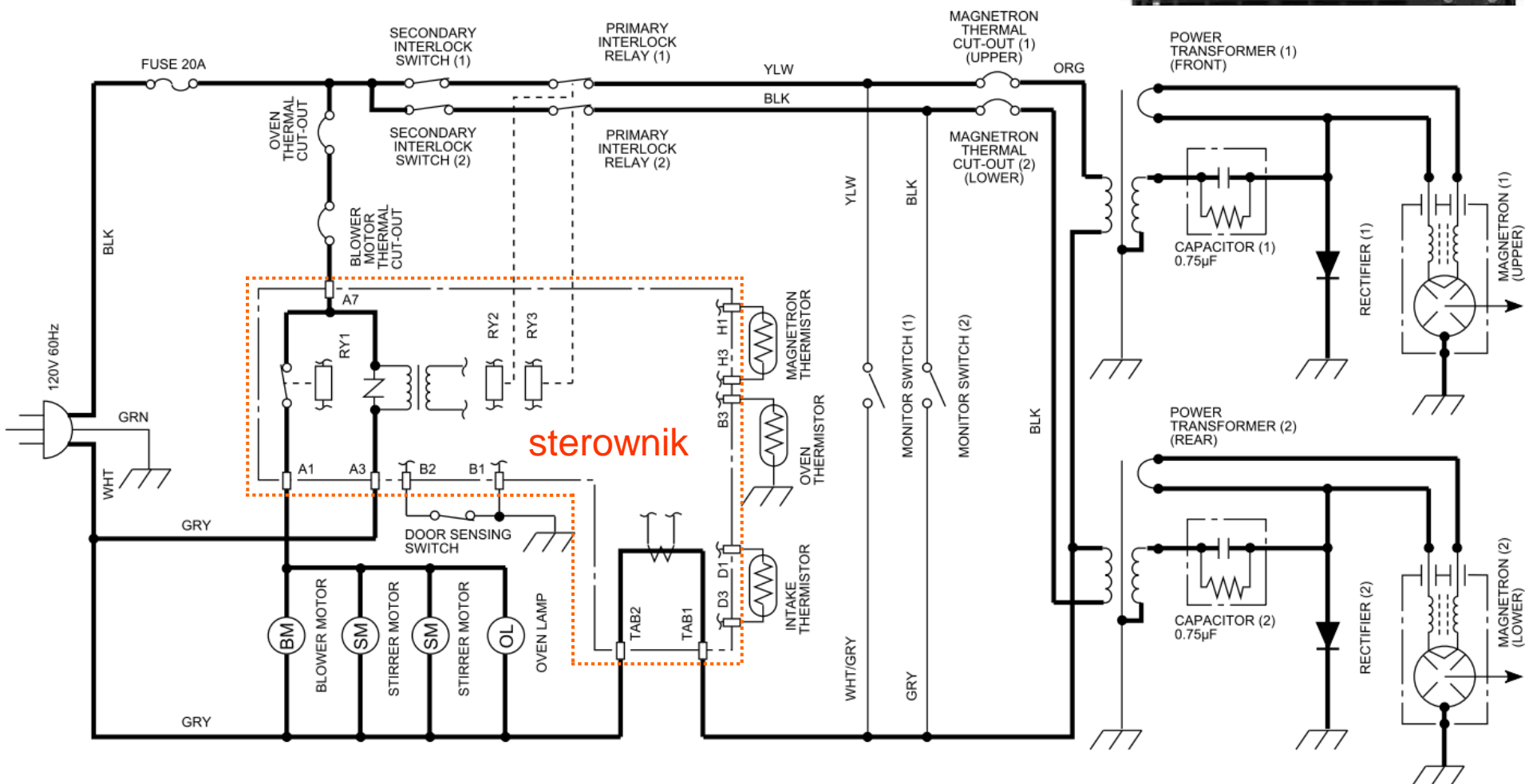
Z reguły *...

- system mikroprocesorowy przeznaczony do sterowania konkretnym urządzeniem (elektromechanicznym, elektronicznym),
- zoptymalizowany (pod względem sprzętowym i programowym) do wykonania ściśle określonych działań (często w określonym czasie),
- wbudowany w urządzenie (stanowi część większej całości),
- wyposażony w zabezpieczenia i układy nadzorujące – umożliwiające w pełni autonomiczną pracę w długim czasie,
- w bardziej skomplikowanych urządzeniach może znajdować się wiele systemów wbudowanych (komunikujących się ze sobą).

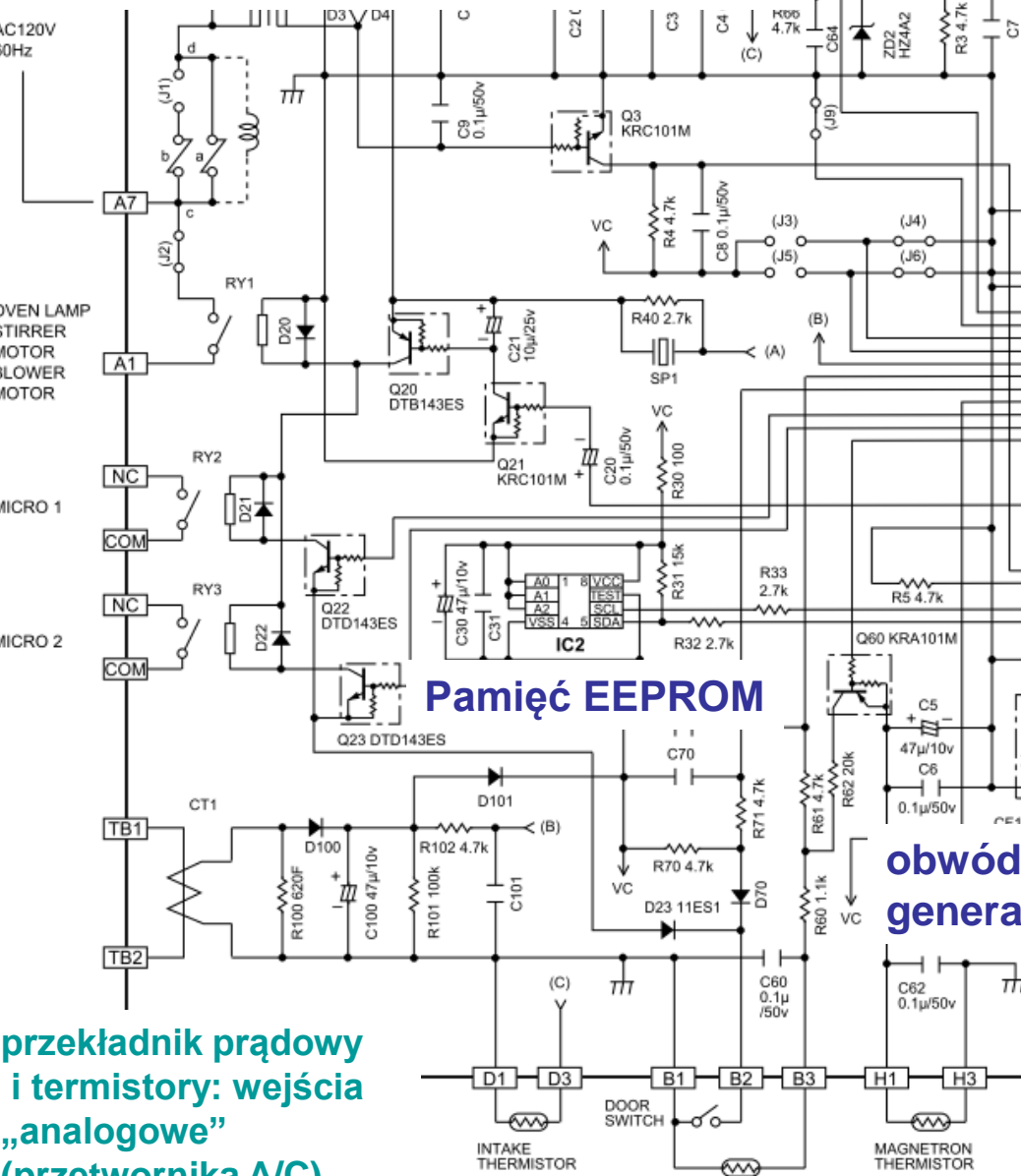
* Rozwój technologii (półprzewodniki, oprogramowanie...) skutecznie rozmywa definicję „systemu wbudowanego”.

Przykłady SW – obecnie prościej wskazać co nie jest systemem wbudowanym...

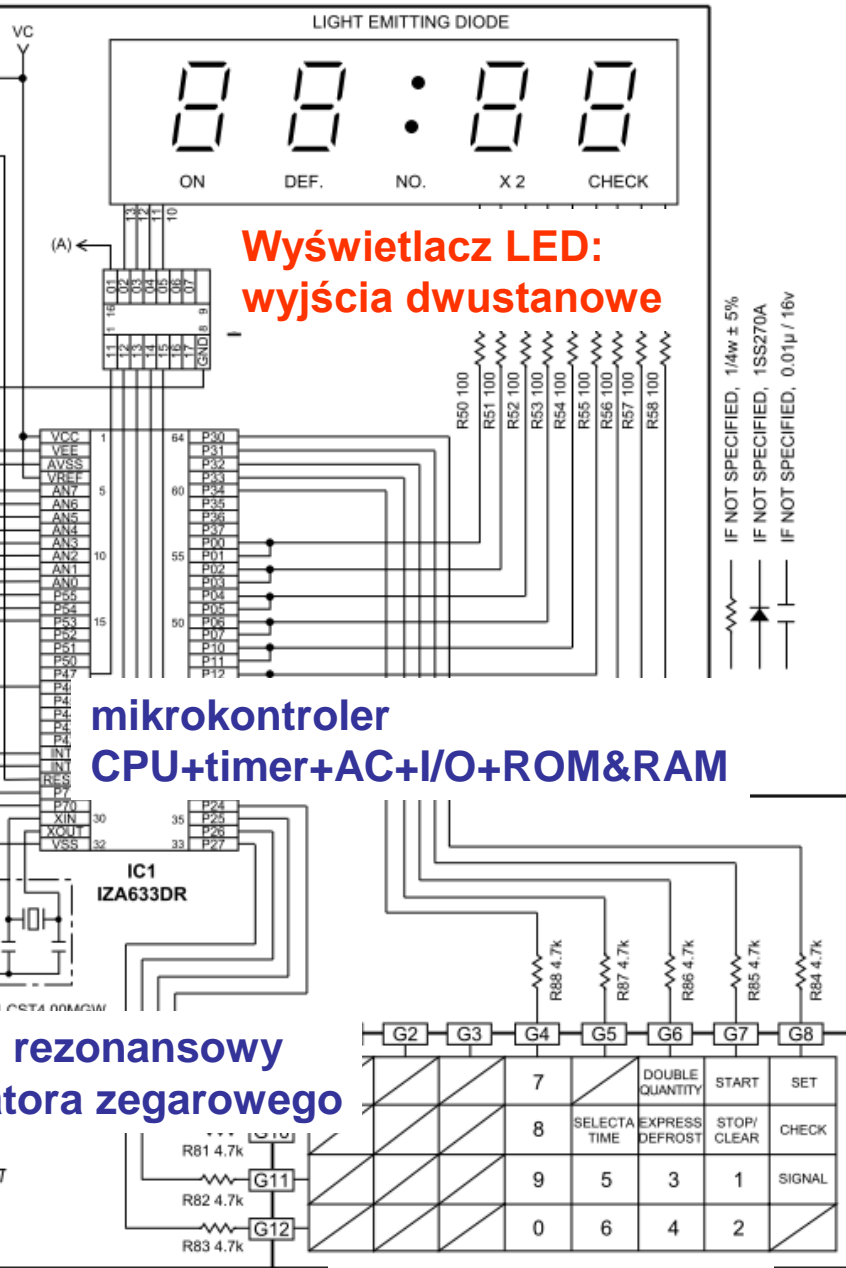
Sterownik kuchenki mikrofalowej R-22
(fragmenty instrukcji serwisowej firmy Sharp)



Sterowanie urządzeniami wykonawczymi dużej mocy: silniki, żarówka, zasilanie magnetronów wyjścia dwustanowe



przekładnik prądowy i termistory: wejścia „analogowe” (przetwornika A/C)



Wyświetlacz LED: wyjścia dwustanowe

mikrokontroler CPU+timer+AC+I/O+ROM&RAM

obwód rezonansowy generatora zegarowego

Klawiatura: wejście (logiczne)

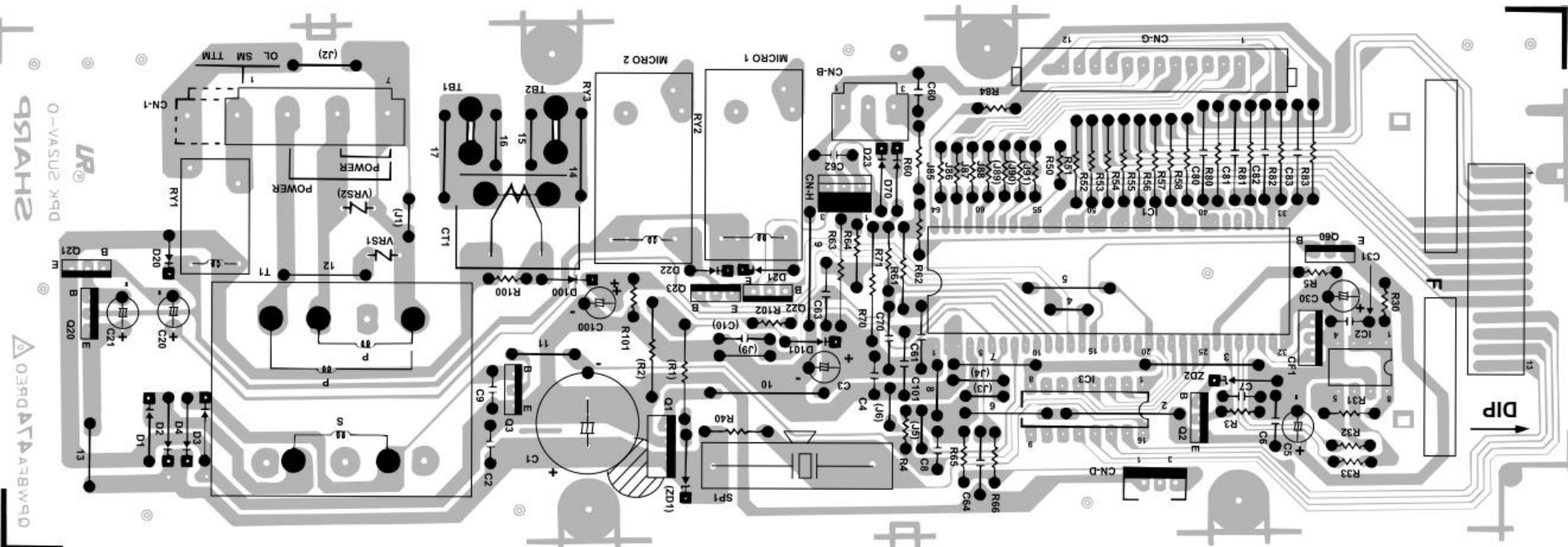
	G2	G3	G4	G5	G6	G7	G8
	/	7	/	DOUBLE QUANTITY	START	SET	
	8	SELECTA TIME	EXPRESS DEFROST	STOP/CLEAR	CHECK		
	9	5	3	1	SIGNAL		
	0	6	4	2			

IF NOT SPECIFIED, 1/4w ± 5%
IF NOT SPECIFIED, 1SS270A
IF NOT SPECIFIED, 0.01µ / 16V

Płytką drukowaną sterownika kuchenki (ang. PCB – Printed Circuit Board):

mozaika ścieżek łączących elementy układu (wg poprzedniego schematu)

wytrawiona w warstwie miedzi pokrywającej laminat (np. z włókna szklanego).



Systemy Wbudowane / Automatyka

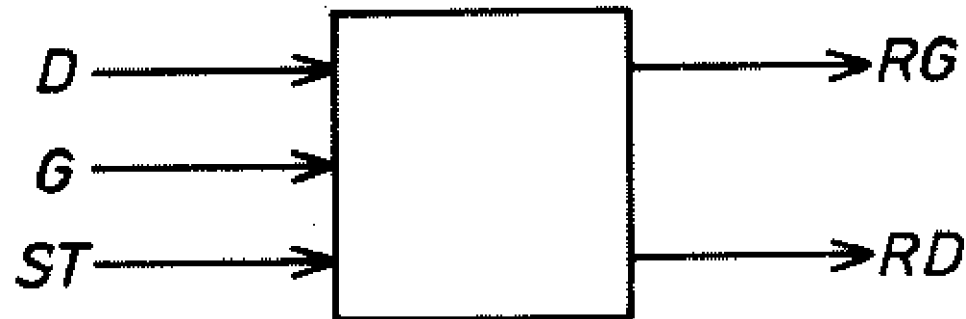
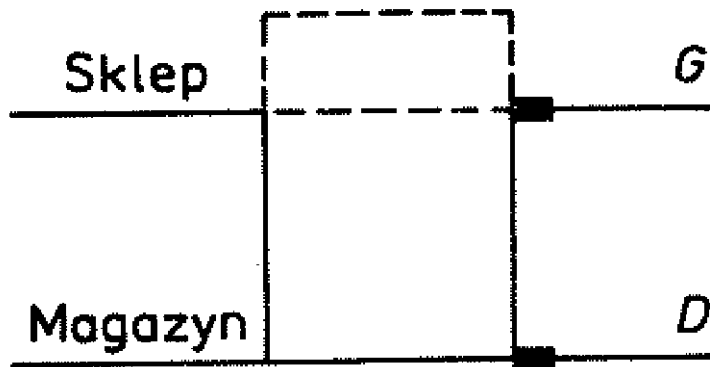
Czym jest sterowanie?

Co to jest sprzężenie zwrotne?

Projekt (bardzo) prostego sterownika - sprzętowego

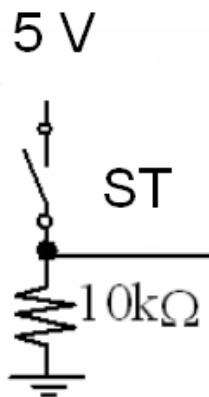
np. windy obsługującej dwie kondygnacje (albo bramy przesuwnej etc.).

- W szybie windy są dwa czujniki położenia końcowego: górnego (G) i dolnego (D).
- Osiągnięcie położenia końcowego sygnalizowane jest stanem wysokim „1”.
- W kabinie znajduje się przycisk START. Jego naciśnięcie (chwilowy stan „1” linii ST) uruchamia windę w kierunku automatycznie ustalonym przez sterownik.
- Zatrzymanie windy następuje automatycznie po osiągnięciu położenia końcowego.
- Kierunek ruchu / obrotów silnika jest ustalany stanem wysokim jednego z sygnałów RG (ruch do góry) lub RD (w dół). Gdy $RG = RD = 0$ winda jest zatrzymana.



cd

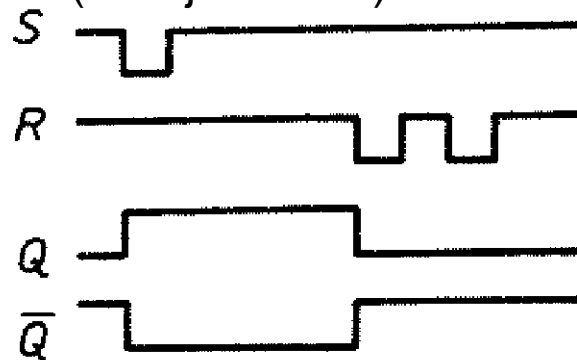
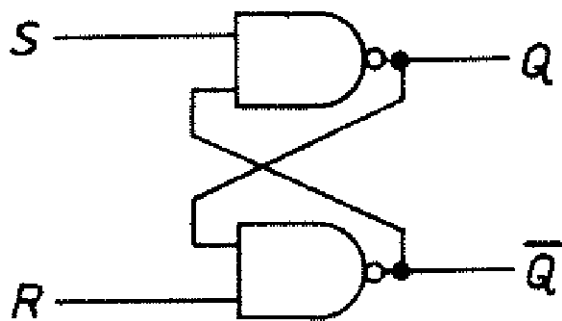
Naciśnięcie przycisku START wmusza **chwilowy** stan wysoki na linii ST (impuls).



Schemat uproszczony,
bez filtracji drgań styków.

Sterownik musi jednak „**zapamiętać**”^{*} polecenie startu i utrzymać włączony silnik aż do osiągnięcia przez kabinę pozycji docelowej.

Potrzebny jest układ pamięciowy – np. przerzutnik (wersja NAND):



S – SET, R – RESET, Q i /Q – wyjścia komplementarne

^{*} to samo dotyczy portu wyjściowego systemu mikroprocesorowego...

cd

Układ sterujący windą może znajdować się w jednym z trzech* stanów:

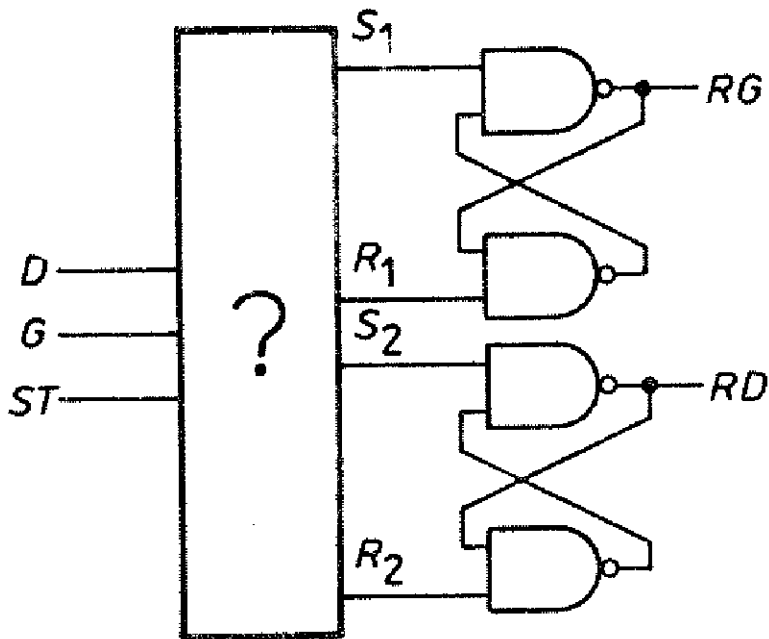
RG=0 i RD=0 – silnik zatrzymany,

RG=0 i RD=1 – jazda w dół,

RG=1 i RD=0 – jazda w górę,

RG=1 i RD=1 – traktujemy jako stan niedozwolony (ew. silnik również zatrzymany).

Potrzebne są dwa przerzutniki ($2^2 =$ cztery możliwe stany):



cd

Reguły:

Przerzutniki (2 x NAND) ustawiane i zerowane są podaniem stanu niskiego (0).

Ruch w górę (RG) powinien się rozpocząć gdy naciśniemy start (ST=1) i kabina jest w położeniu dolnym (D=1)

$$S_1 = \overline{D \cdot ST}$$

i trwać aż do osiągnięcia górnego położenia końcowego (G=1):

$$R_1 = \overline{G}$$

Ruch w dół (RD) powinien się rozpocząć gdy naciśniemy start (ST=1) i kabina jest w położeniu górnym (G=1)

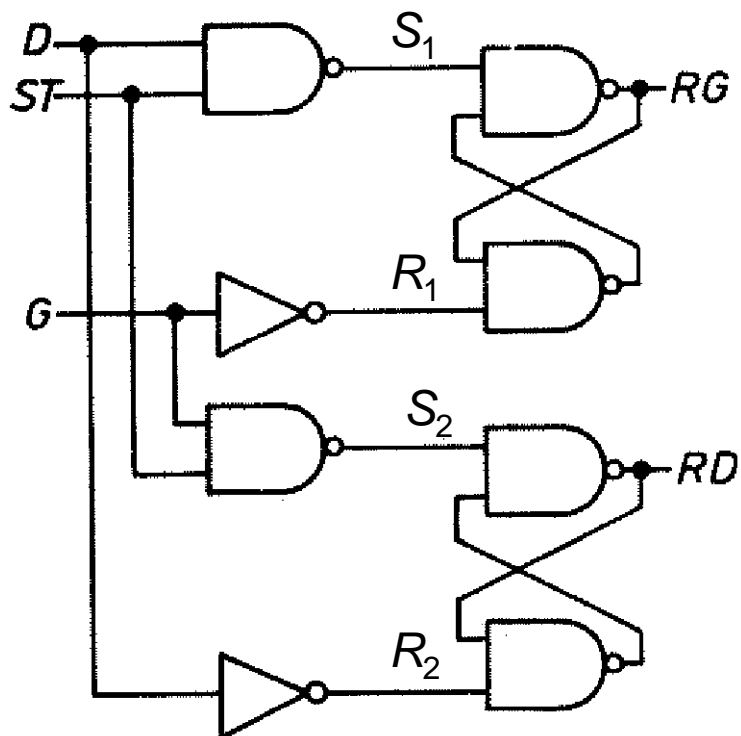
$$S_2 = \overline{G \cdot ST}$$

i trwać aż do osiągnięcia dolnego położenia końcowego (D=1):

$$R_2 = \overline{D}$$

cd

Przykładowy schemat całego układu:



Możliwe sposoby wykonania sterownika windy:

- elementy dyskretne (kilkanaście tranzystorów i oporników, kilka diód) 5 x 5 mm
 - dwa układy 7400 (każdy zawiera cztery bramki NAND) 10 mm
 - jeden układ PLD (np. typu 16V8)
 - prosty mikrokontroler 8bit np. ATTINY25 6.4 x 25 mm
- 
- 
- 

Jakie kryteria brać pod uwagę ?

- gabaryty
 - komplikacja układu
 - robocizna / koszt montażu
 - koszt elementów / programu
 - możliwość wprowadzania zmian
 - szybkość działania, czas reakcji, moc obliczeniowa
 - ochrona przed skopiowaniem
 - pobór energii (ew. wydzielanie ciepła)
 - odporność na warunki atm. (i inne zakłócenia np. promieniowanie jonizujące)
 - inne...
- 

Programowalne układy logiczne

(C)PLD – (Complex) Programmable Logic Device

FPGA – Field Programmable Gate Array

Konfigurowalne przy pomocy języków opisu sprzętu
(Hardware Description Language) np.:

CUPL

VHDL

Verilog

projekt na bazie schematu logicznego

PLD:

CUPL – wyrażenia logiczne:

/romwr = /WR + A13 + A14 + A15 + rom + /cm

Verilog

Licznik mod 16

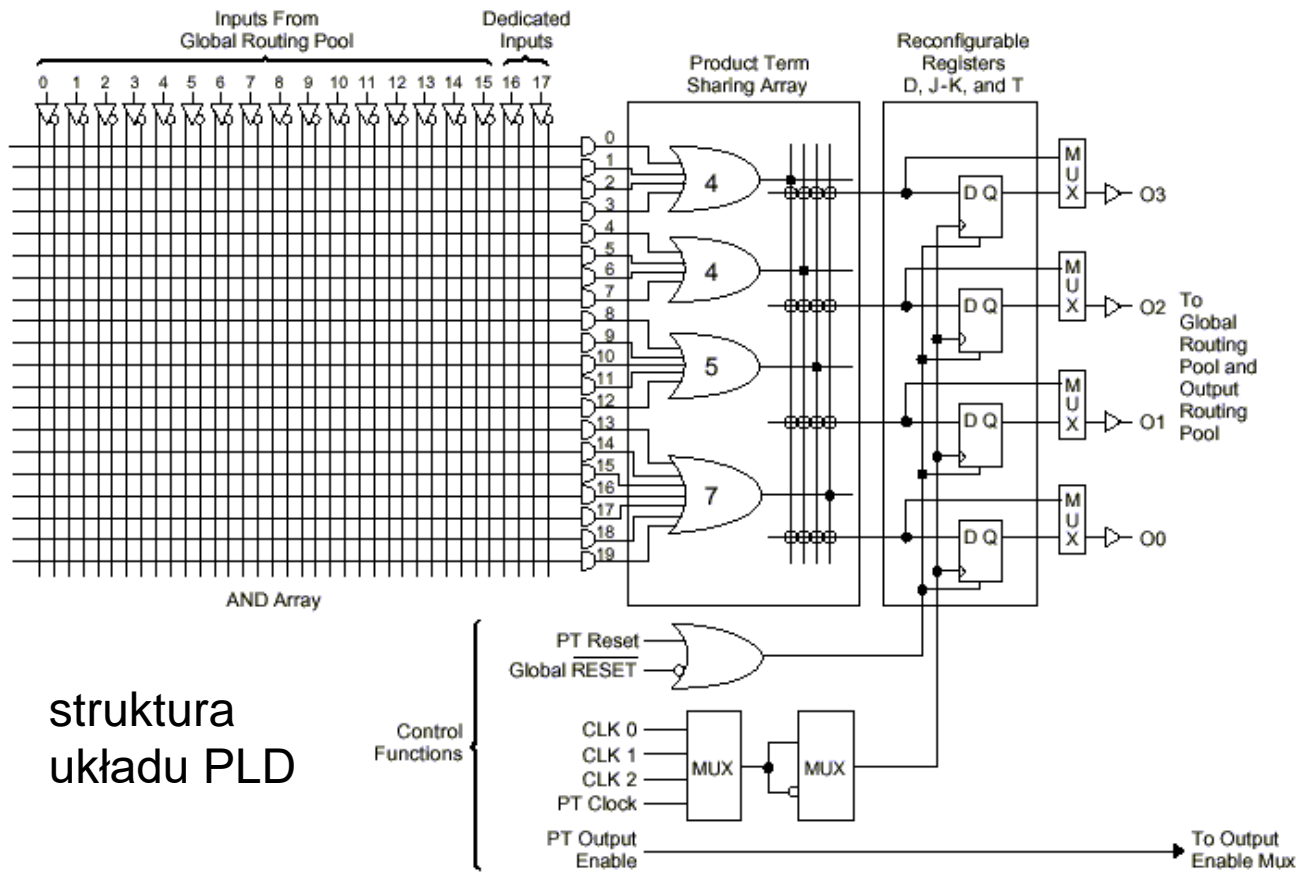
```

module counter(CLOCK, DIRECTION, COUNT_OUT);
input CLOCK;
input DIRECTION;
output [3:0] COUNT_OUT;
);

reg [3:0] count_int = 0;
always @(posedge CLOCK)
if (DIRECTION)
count_int <= count_int + 1;
else
count_int <= count_int - 1;

assign COUNT_OUT = count_int;
endmodule

```

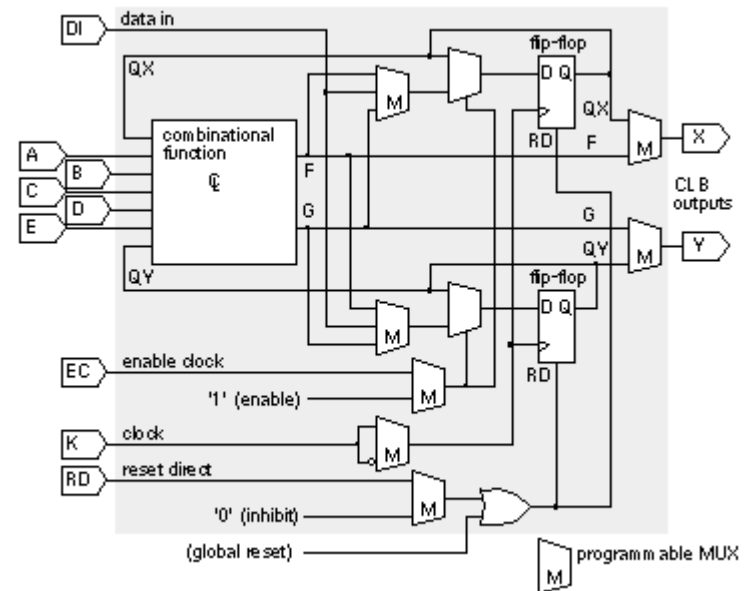
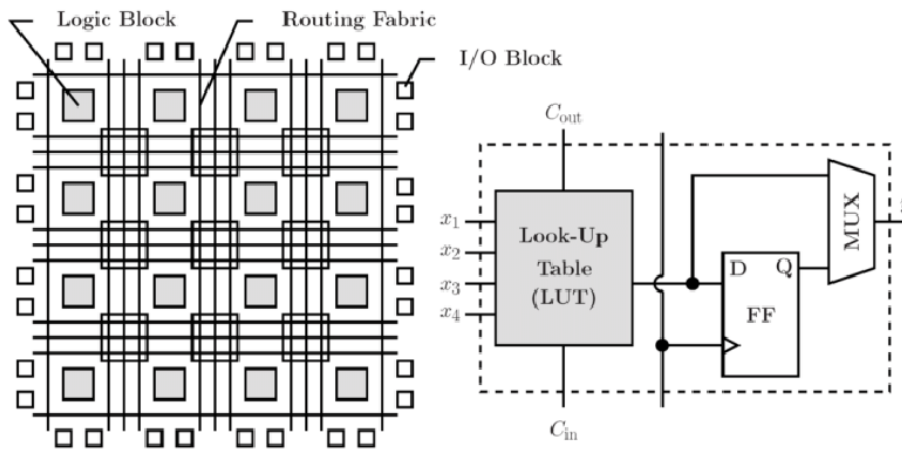


struktura układu PLD

FPGA:



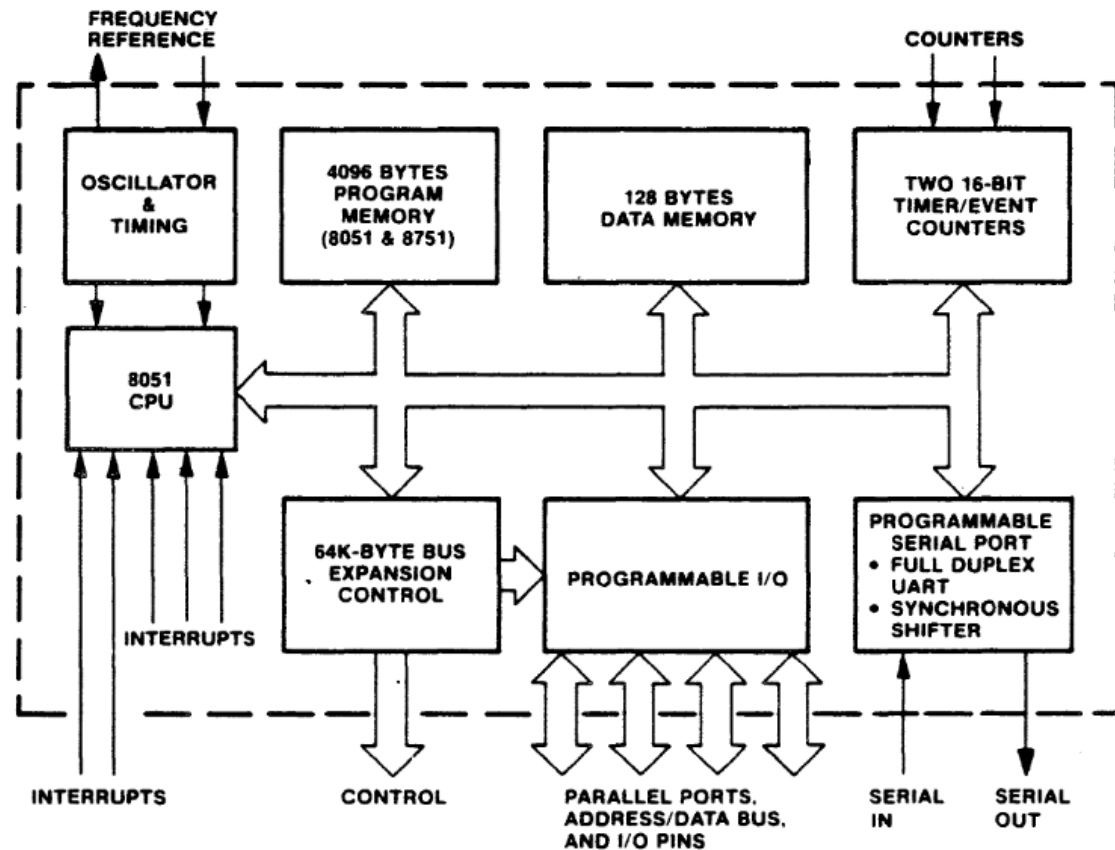
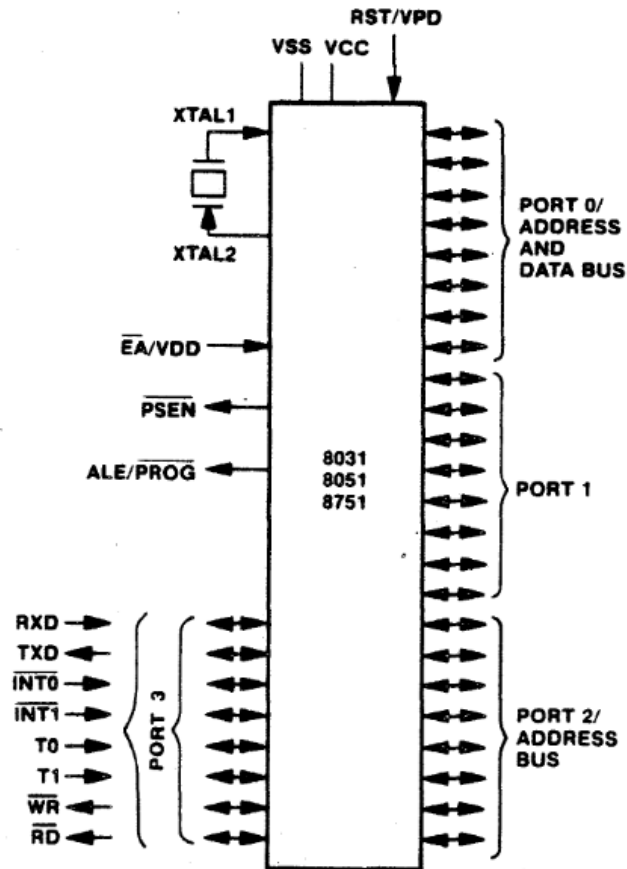
FPGA logic cells



Mikrokontrolery

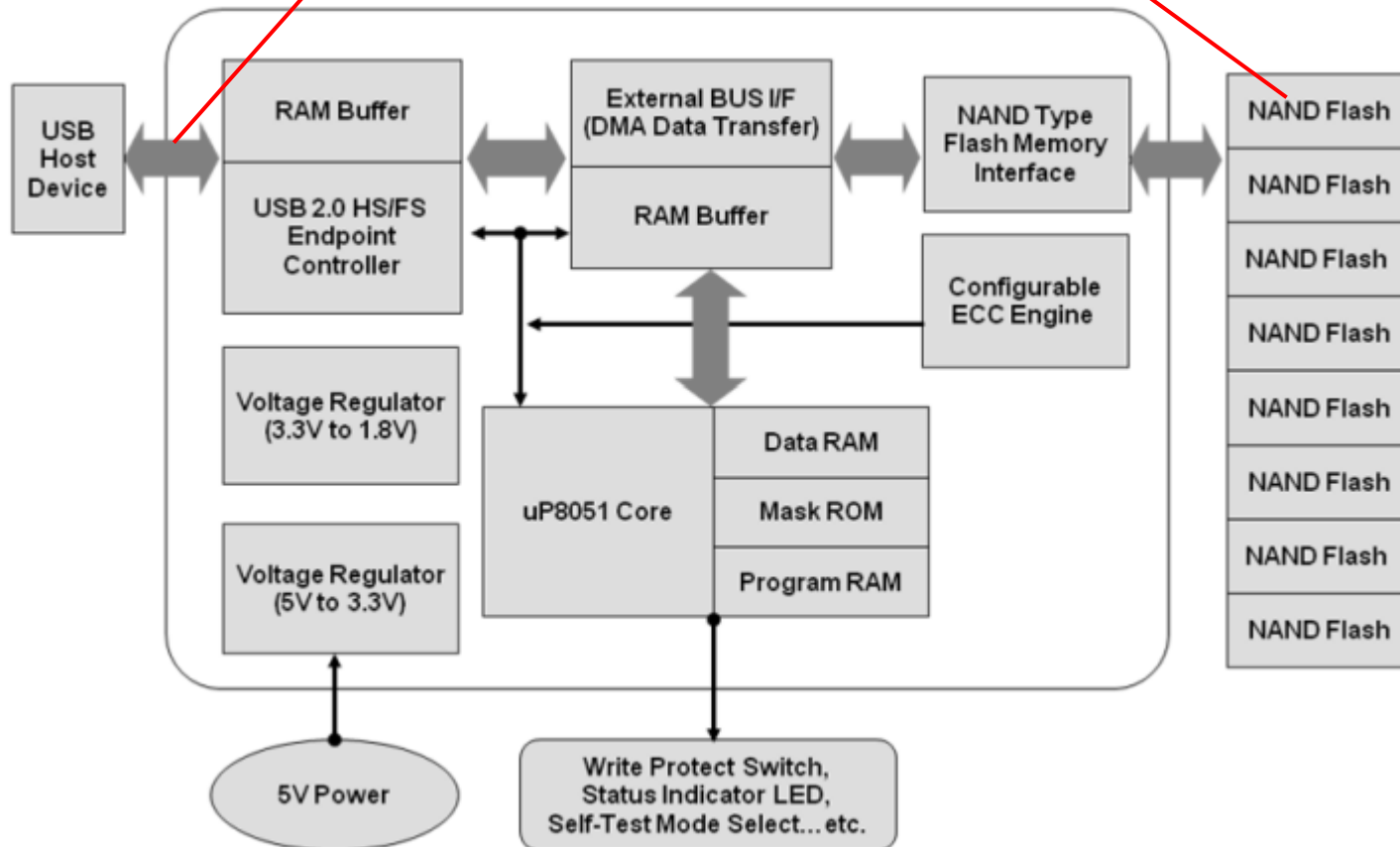
8051 (8-bit, CISC)

0.5 – 1 MIPS @ 12 MHz



Przykład „wbudowanego” rdzenia 8051 – pendrive – kontroler Flash <-> USB

układ SM3257EN
(Silicon Motion)



analogicznie: kontrolery myszy komputerowej, klawiatury itp.

Mikrokontrolery AVR

8-bit RISC

6-20 MIPS @ 20 MHz

(instrukcje 1-3 cyklowe)

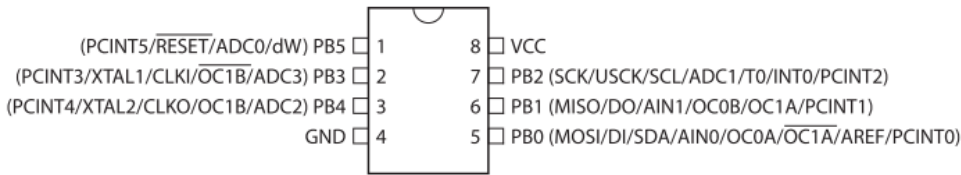
zintegrowany 10bit ADC

ATTINY25/45/85

2/4/8 kB Flash ROM (program)

128-512 B RAM

PDIP/SOIC/TSSOP

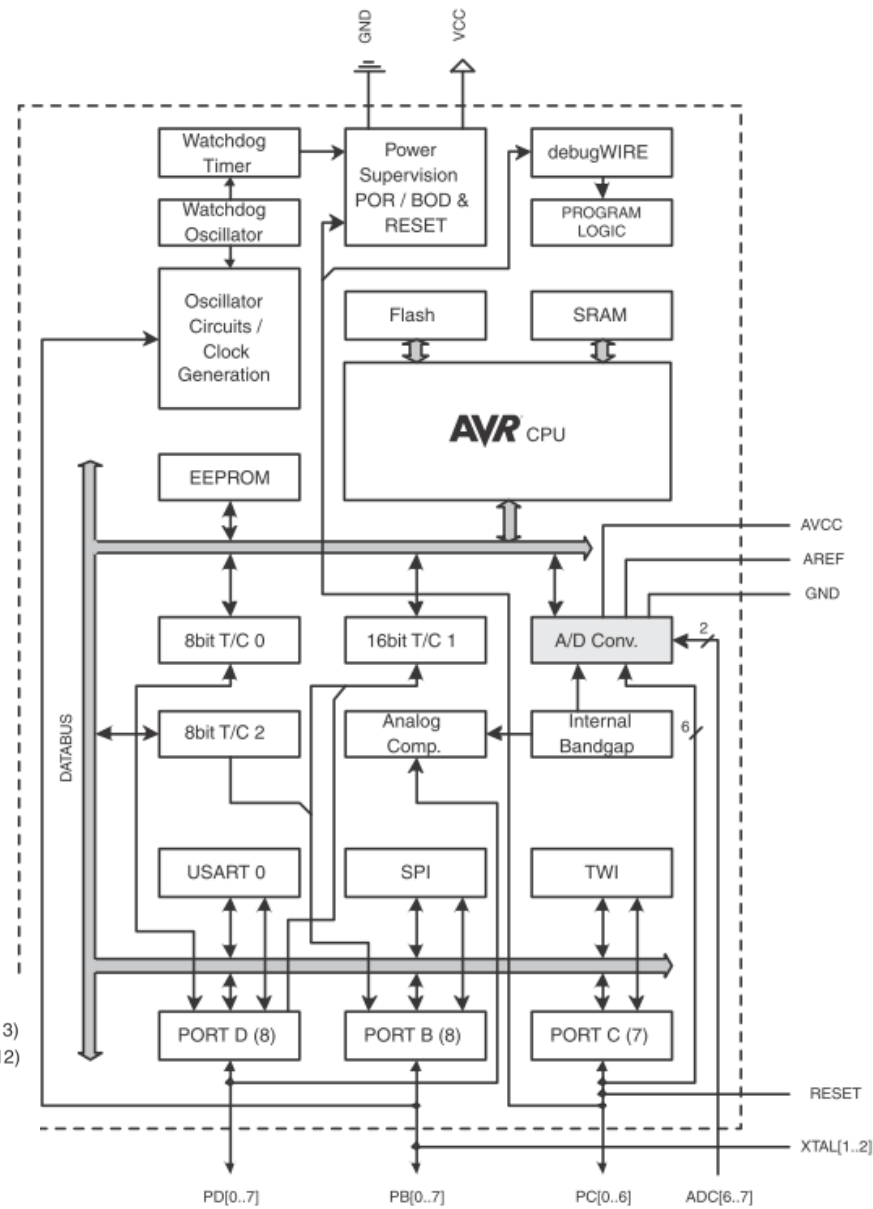
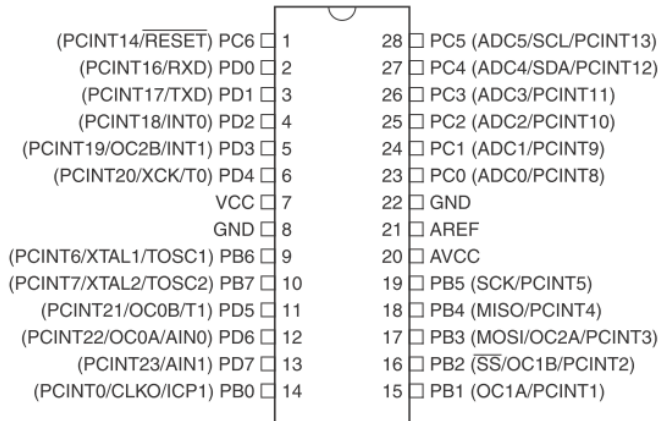


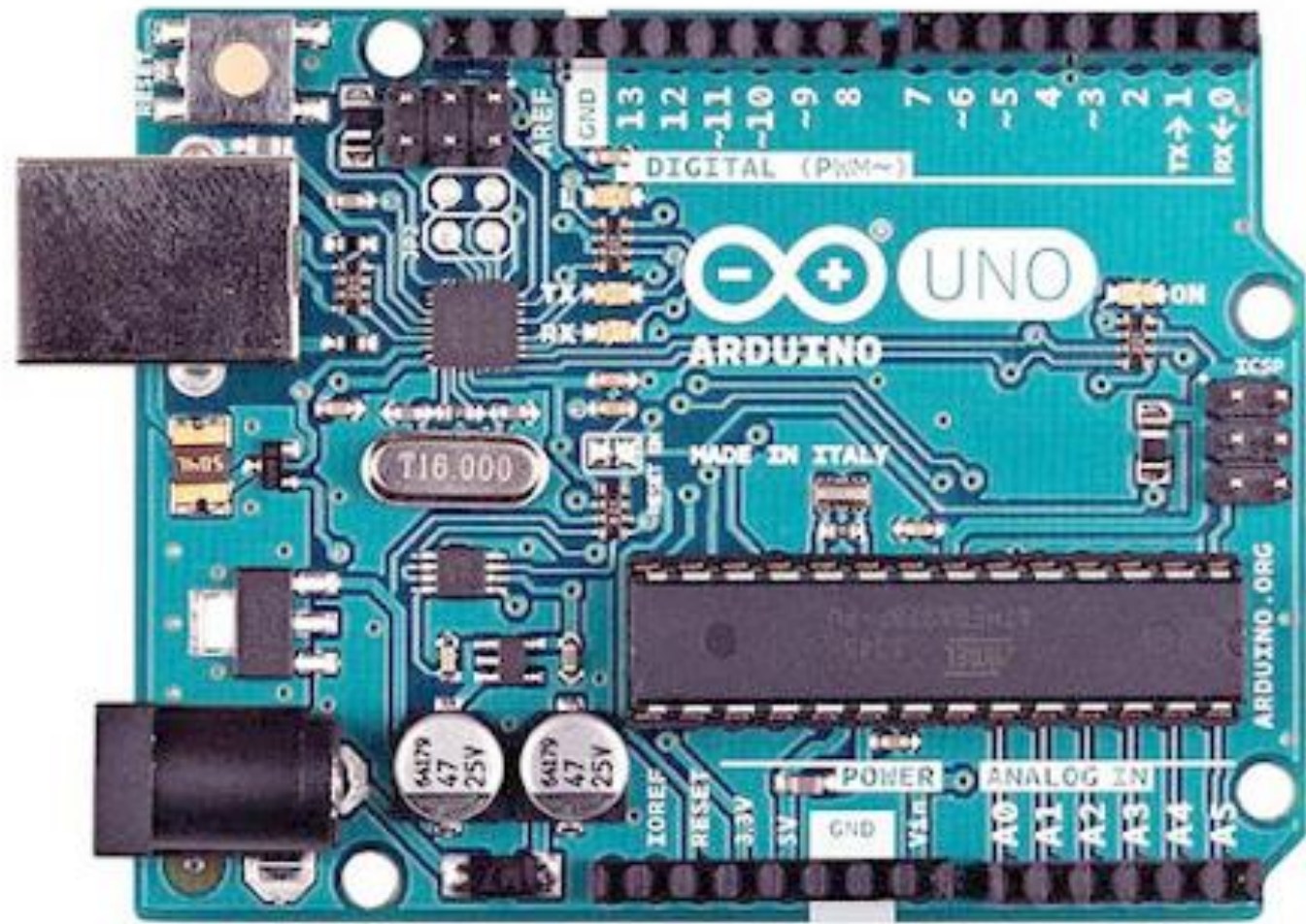
ATMEGA 328

(np. użyta w Arduino)

32 kB Flash ROM

512 B – 2 kB RAM





ARDUINO UNO

MADE IN ITALY

ARDUINO.ORG

DIGITAL (PWM~)
13 GND
12
11
10
9
8
7
6
5
4
3
2
1 TX →
0 RX ←

IOREF
RESET
3.3V
5V
GND
VCC
ANALOG IN
A0
A1
A2
A3
A4
A5

T16.000

6AL79
47
25V

6AL79
47
25V

ATMEGA328P

RELAY

ON

ICSP

Mikrokontrolery ARM-Cortex M4


32-bit RISC

Np. rodzina STM32 (producent: ST)

- o nich szerzej na kolejnych wykładach i laboratoriach

Cortex®-M4 (DSP + FPU) – Up to 180 MHz

- ART Accelerator™ enabling 0 wait state executing from internal Flash
- Up to 2x USB2.0 OTG FS/HS
- SDIO
- USART, SPI, PC
- PS + audio PLL
- 16 and 32-bit timers
- Up to 3x 12-bit ADC (0.41 μs)
- Up to 2x 12-bit DAC
- External memory controller (except for STM32F411/STM32F401)
- Low voltage 1.7¹ to 3.6 V

 Product	F _{cpu} (MHz)	Flash (bytes)	RAM (KB)	Ethernet I/F IEEE 1588	Camera I/F	SDRAM I/F		SAP ³ I/F	Chrom-ART Graphic Accelerator™	TFT LCD controller	MPI DSI
				2x CAN		Dual SPI	Quad SPI	SPDIF RX			
STM32F469 ²	180	512 K to 2 M	384	•	•	•	•	•	•	•	•
STM32F429 ²	180	512 K to 2 M	256	•	•	•	•	•	•	•	
STM32F427 ²	180	1 to 2 M	256	•	•	•	•	•	•		
STM32F446	180	256 K to 512 K	128	•	•	•	•	•			
STM32F407 ²	188	512 K to 1 M	192	•	•						
STM32F405 ²	188	512 K to 1 M	192	•							

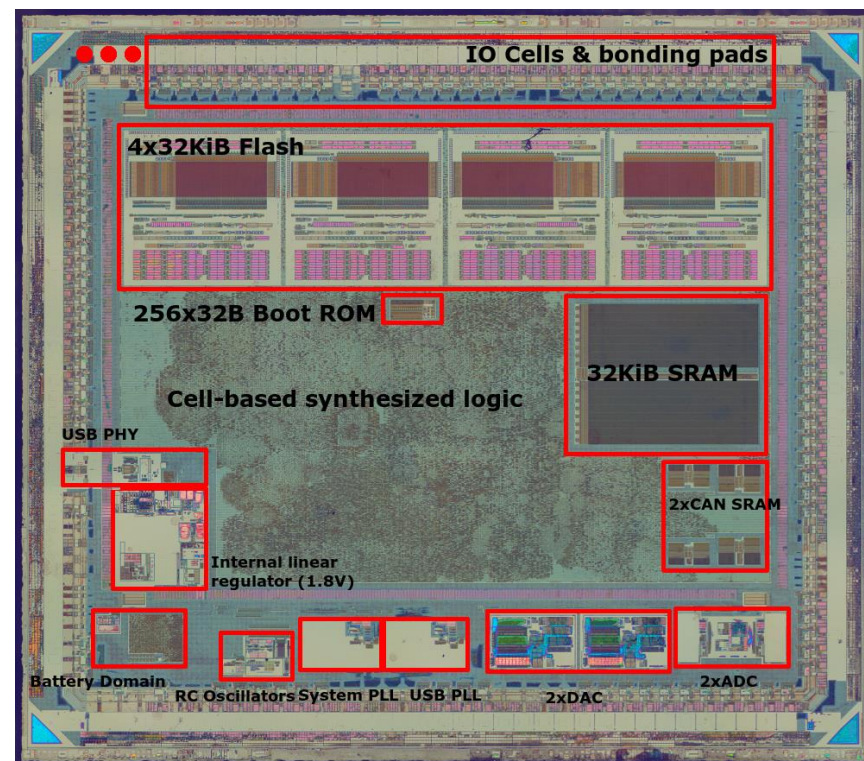
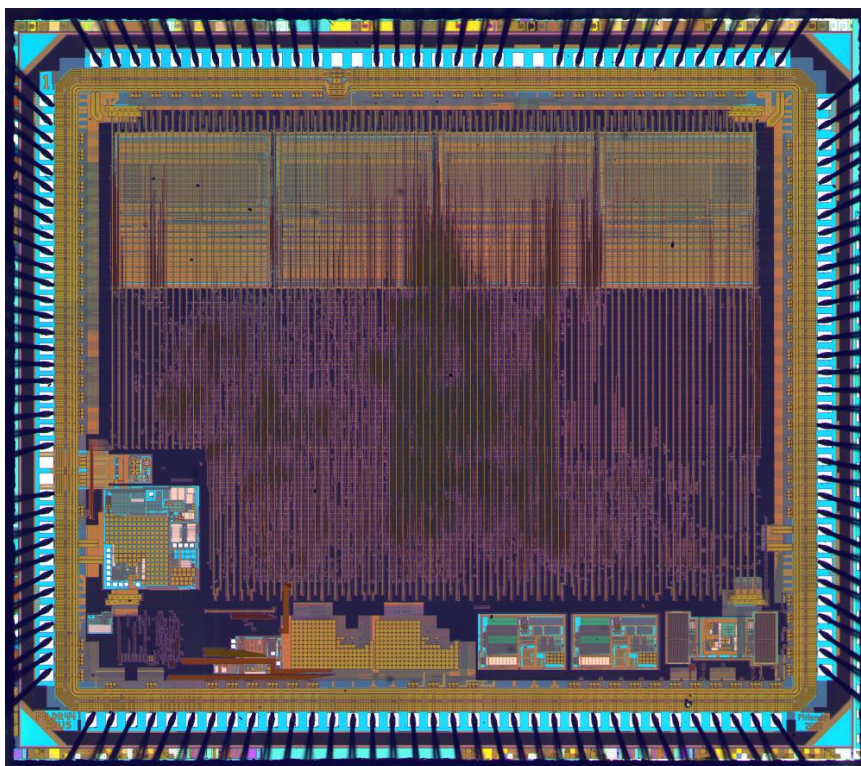
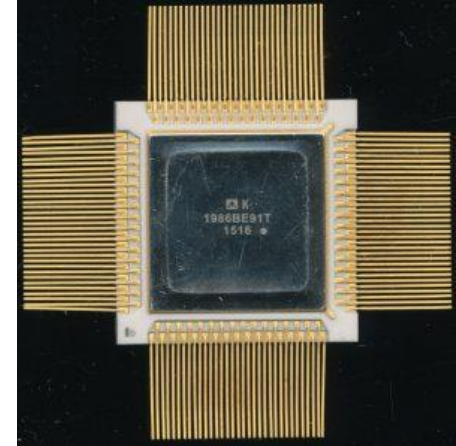
Product	F _{cpu} (MHz)	Flash (KB)	RAM (KB)	Dynamic Efficiency™	RUN current (μA/MHz)	STOP current (μA)	Small package (mm)	DMA Batch Acquisition Mode
STM32F411	100	256 to 512	128	•	Down to 100	Down to 12	Down to 3.034x3.22	•
STM32F401	84	128 to 512	96	•	Down to 128	Down to 10	Down to 3x3	

Notes: 1/ 1.7 V min on specific packages

2/ The same devices are also found with embedded Hardware crypto/hash

3/ Serial Audio I/F

Mikrokontroler 1986VE91T (ARM Cortex M3), proces 180nm (prod. Milandr, Rosja)



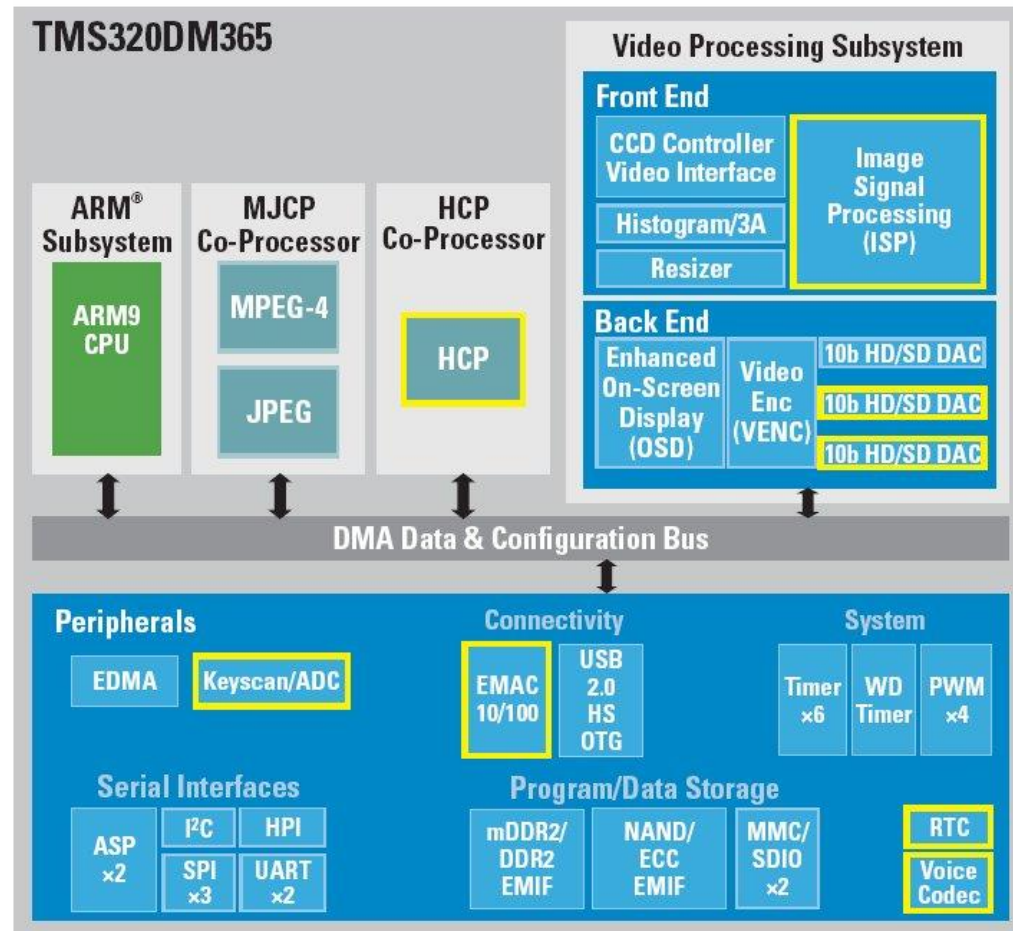
źródło:

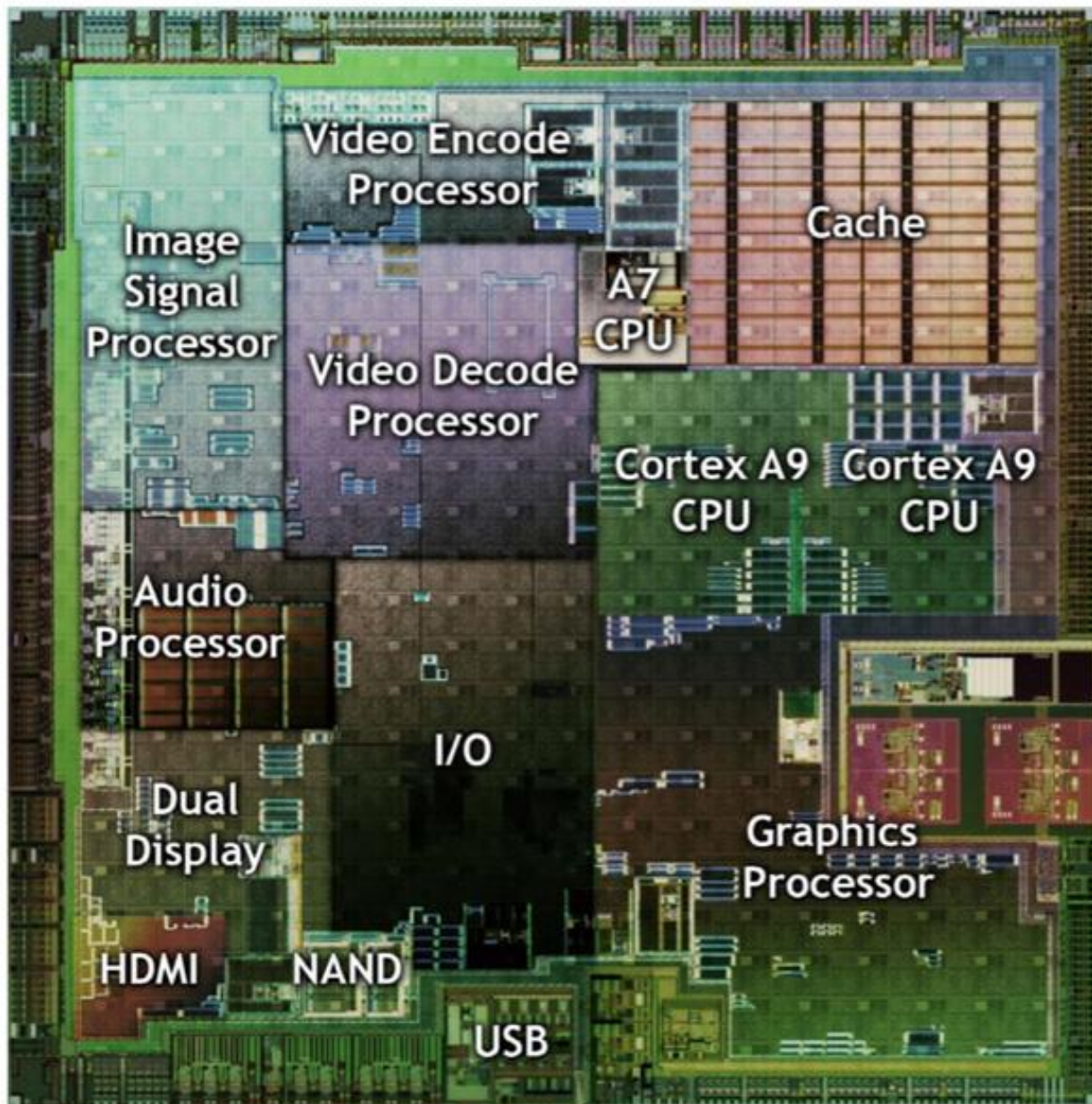
zeptobars.com/en/read/MDR32F9Q2I-1986VE91T-whats-inside-russian-arm

System on Chip

Podobnie jak w przypadku mikrokontrolerów większość bloków niezbędnych do funkcjonowania systemu zawartych jest w jednej strukturze krzemowej (również przetworniki A/D D/A, układy analogowe, komunikacji radiowej itp.)

W przeciwieństwie do mikrokontrolera, procesor(y) (również GPU / DSP) w SoC są bardziej złożone, dużej mocy obliczeniowej, również wielordzeniowe.





Przykład SoC

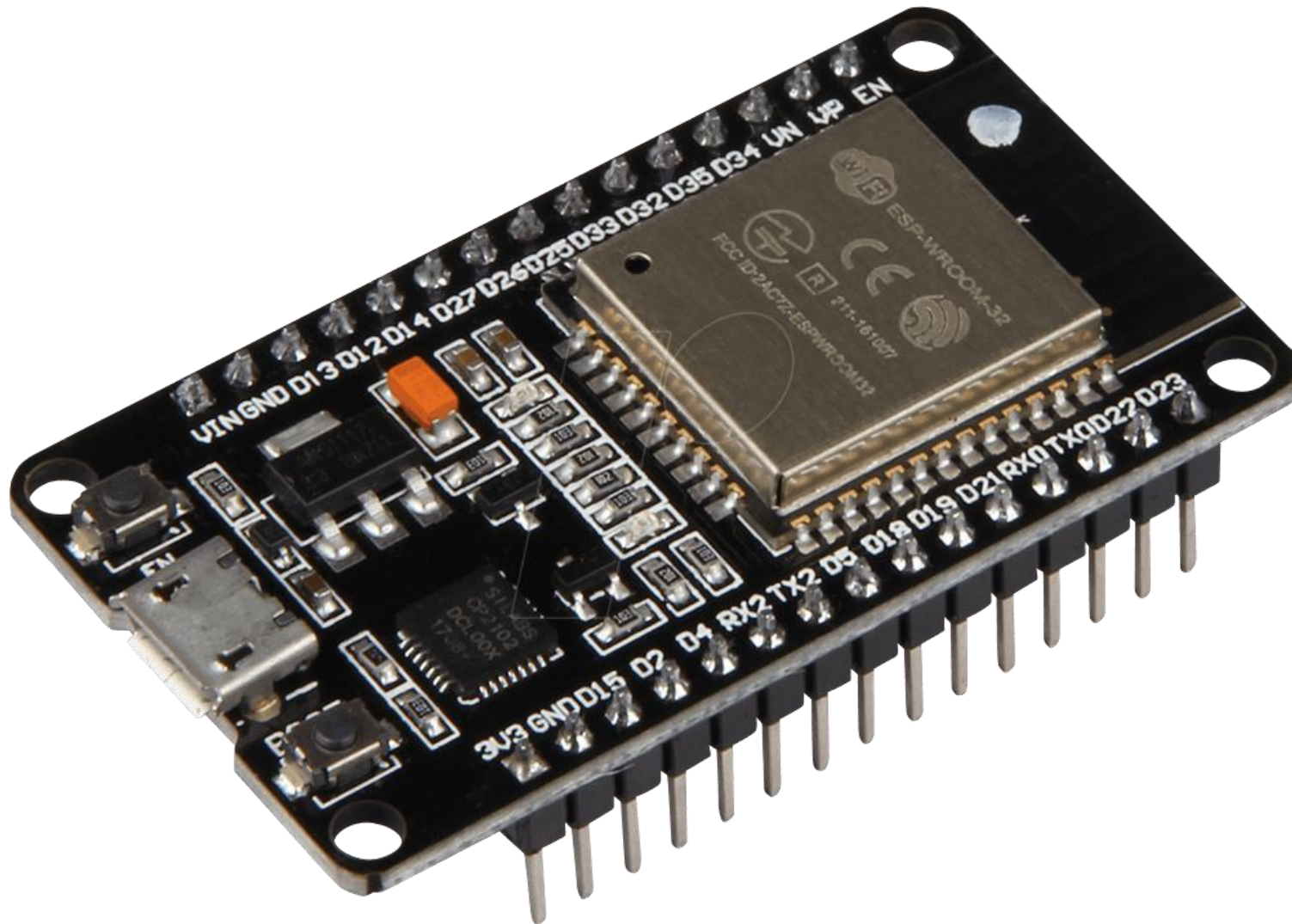
NVIDIA/Qualcomm Tegra2
(Smartphone, PDA itp.)

i kolejne wersje:

Tegra X: 1 4 x Cortex A53 4 x Cortex A57, Xavier...

Przykład SoC

Espressif ESP-32



Przykład SoC

Zdjęcia z:

electronupdate.blogspot.com/2018/08/espressif-esp32-teardown.html

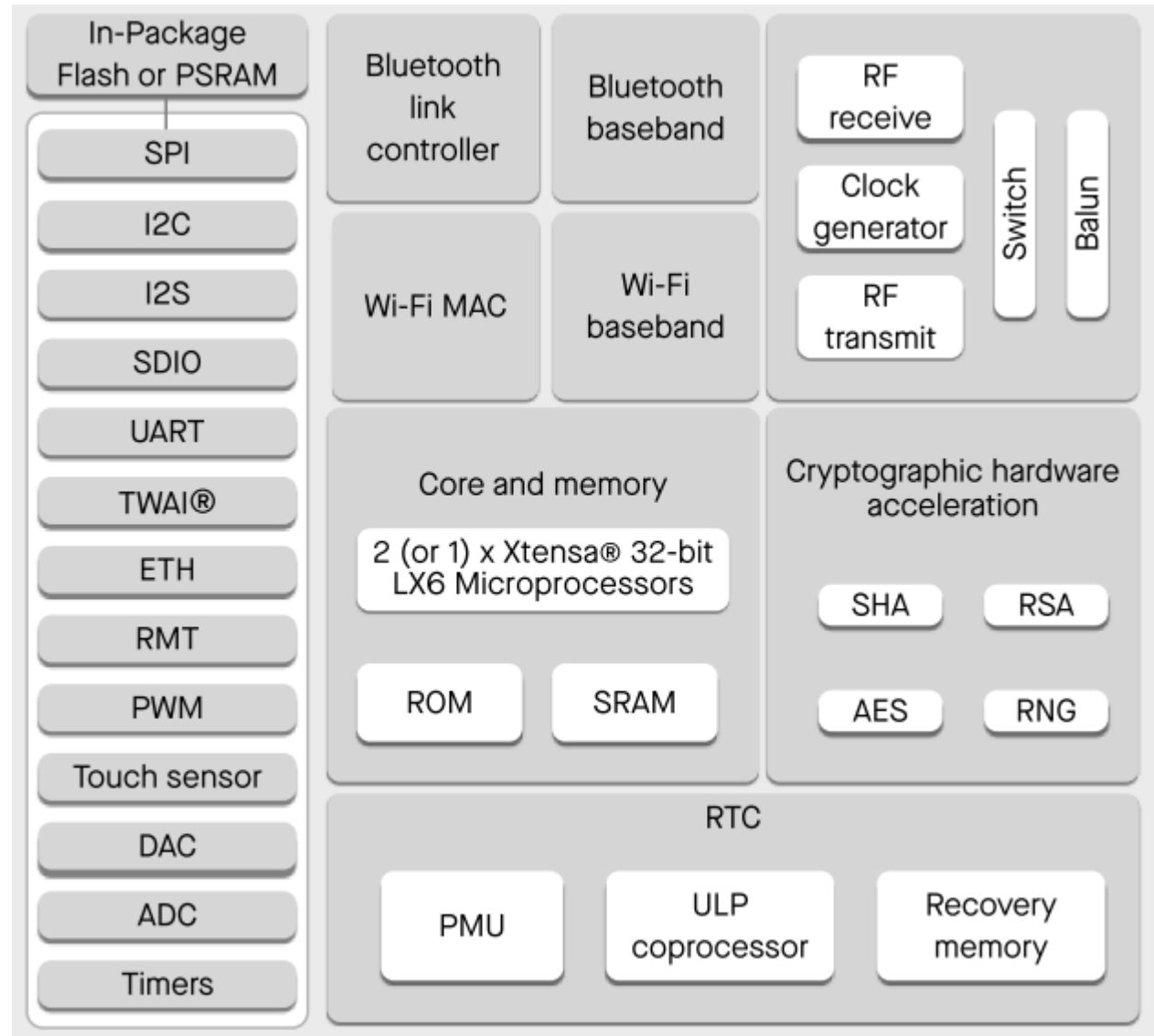
zeptobars.com/en/read/Espressif-ESP32-Wi-Fi-Bluetooth-2.4Ghz-ISM

Espressif ESP-32



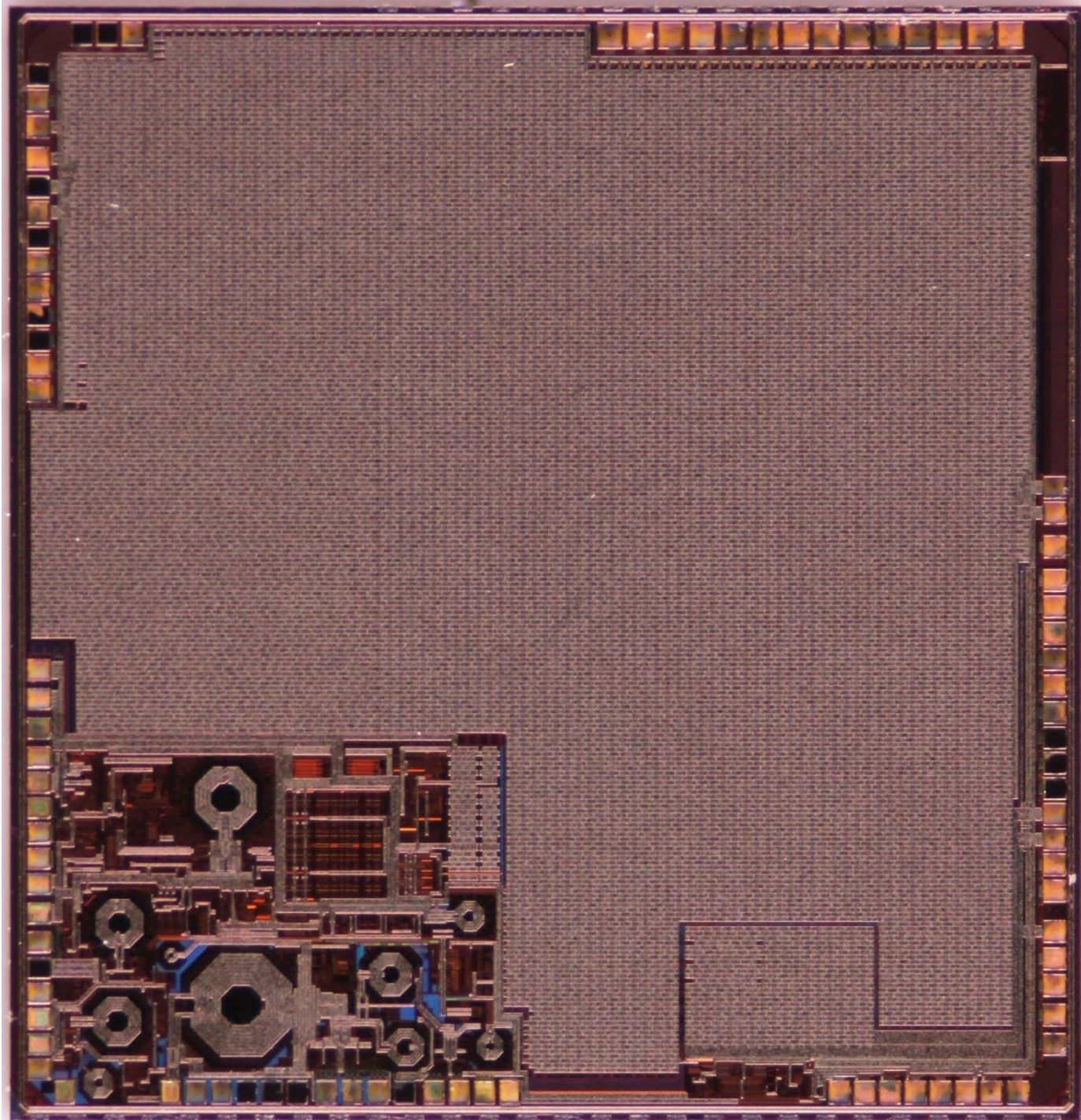
Przykład SoC

Espressif ESP-32



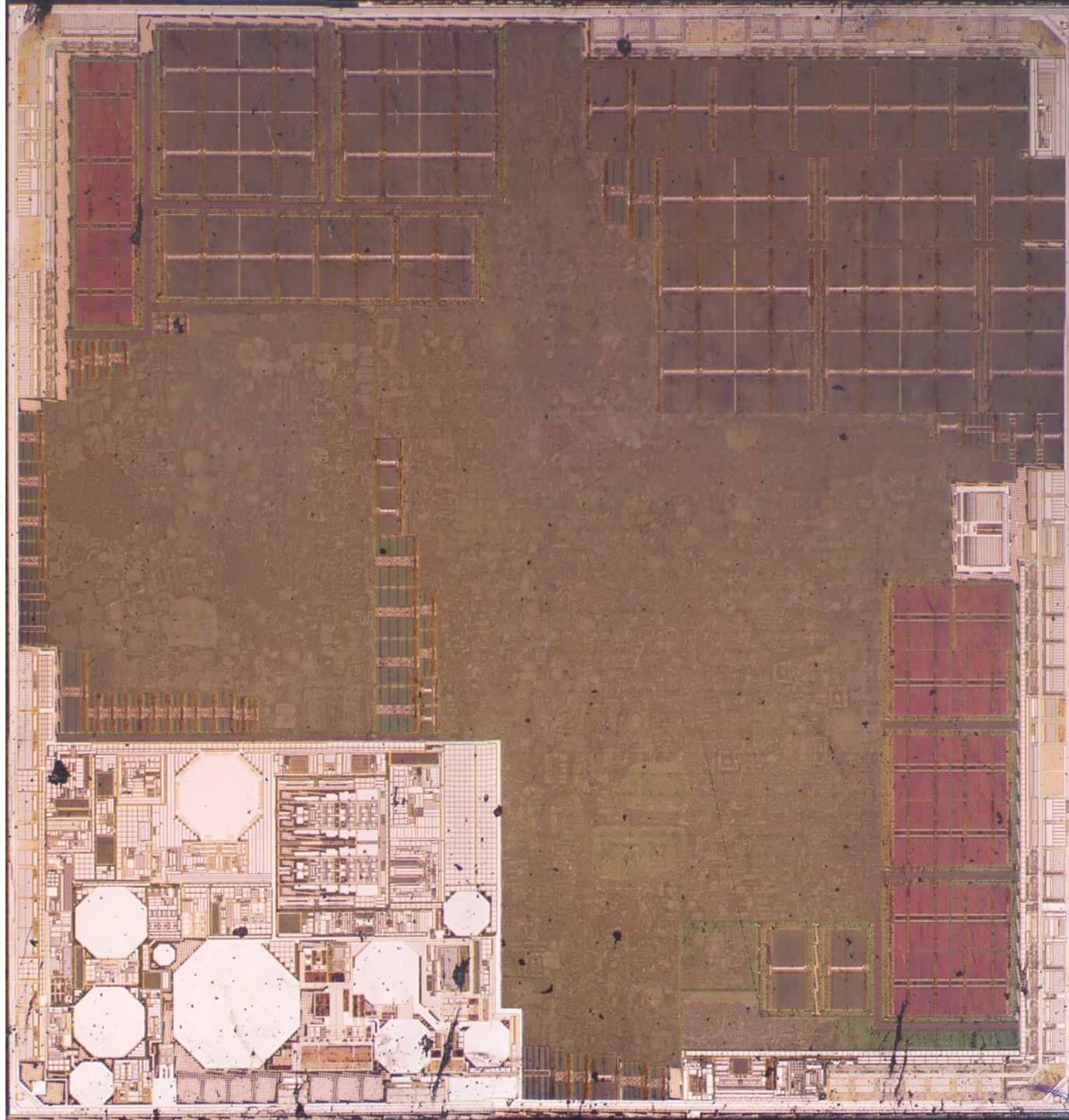
Przykład SoC

Espressif ESP-32

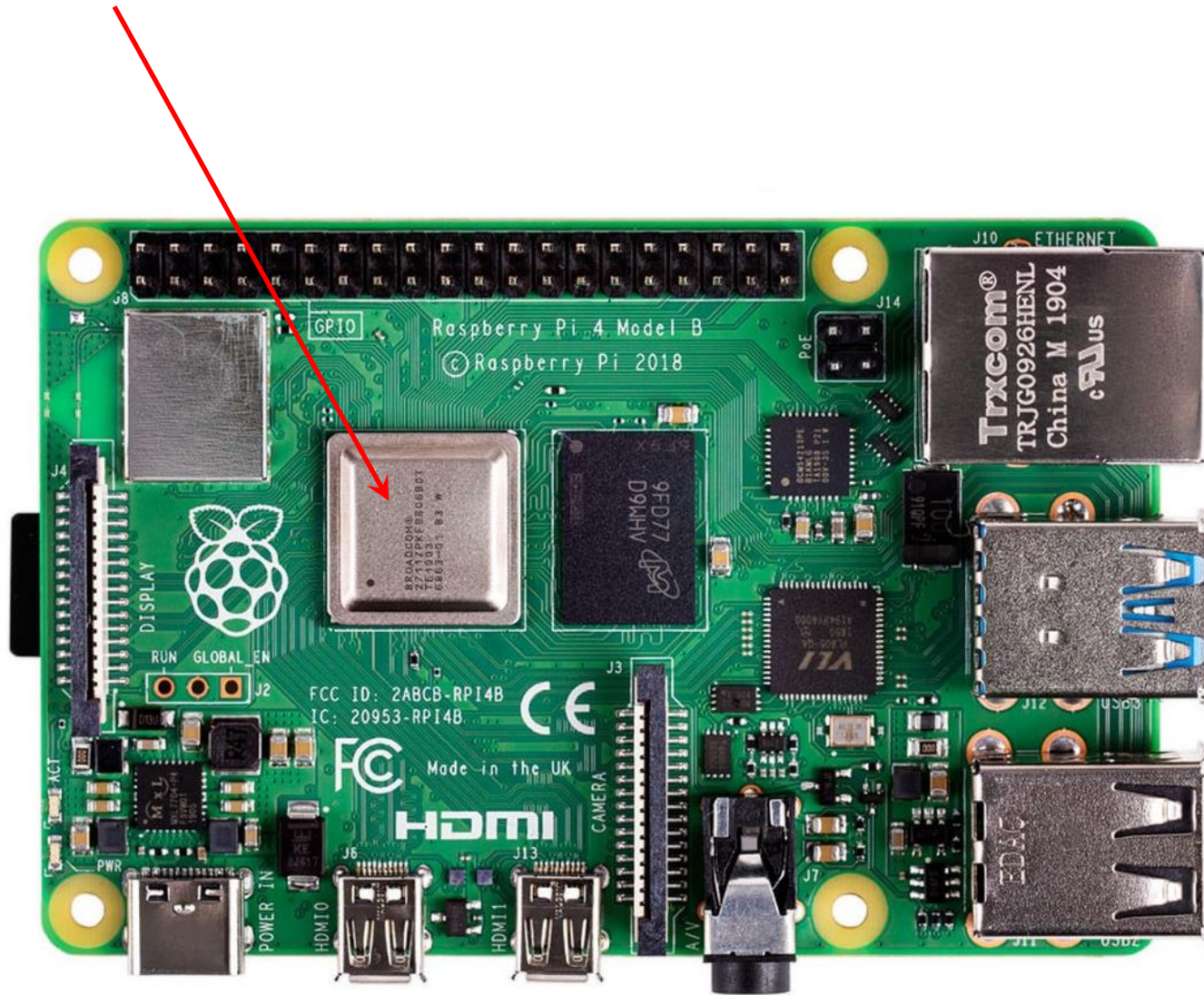


Przykład SoC

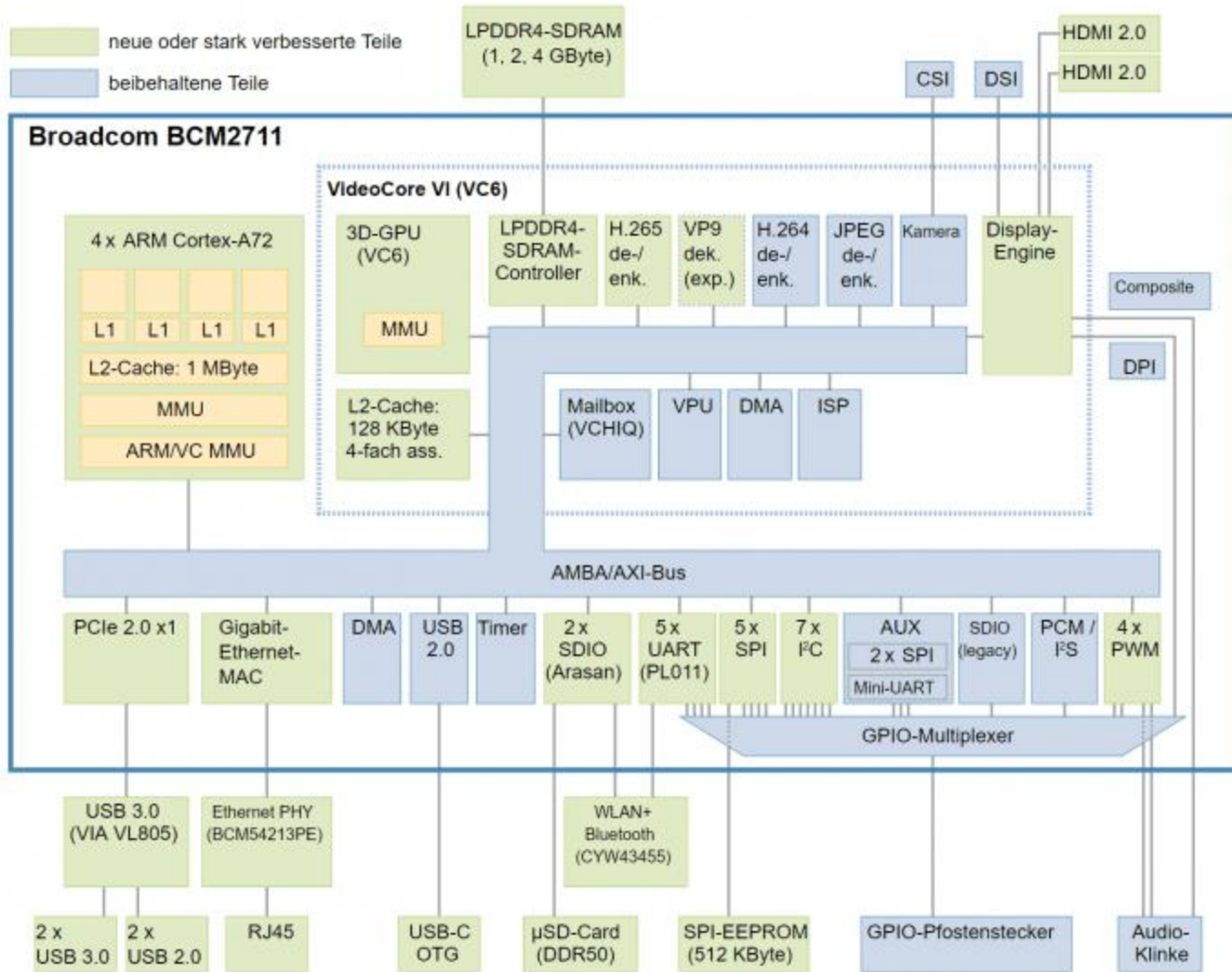
Espressif ESP-32



Broadcom BCM2711 (SoC w Raspberry Pi 4b)



Broadcom BCM2711 (SoC w Raspberry PI 4b)

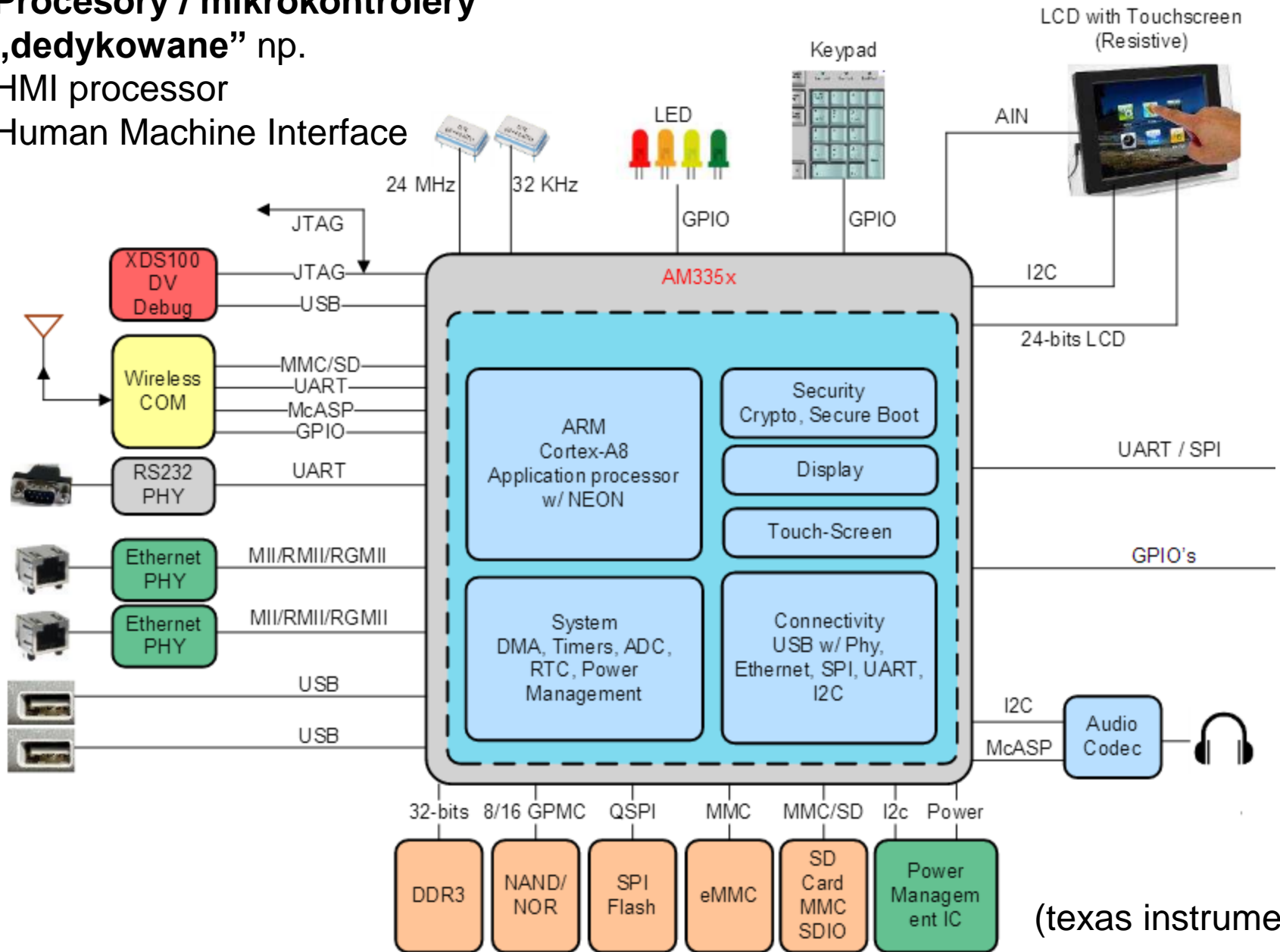


Processor / mikrokontrolery

„dedykowane” np.

HMI processor

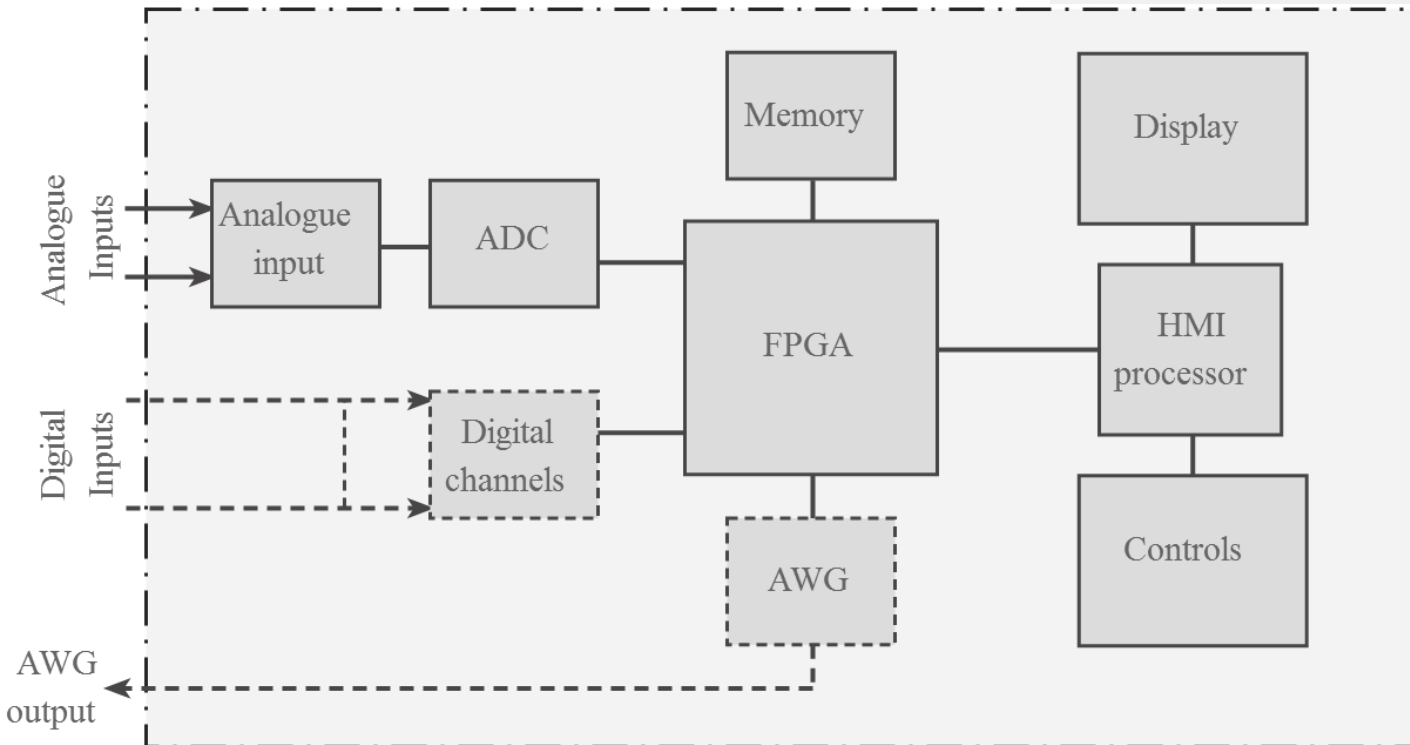
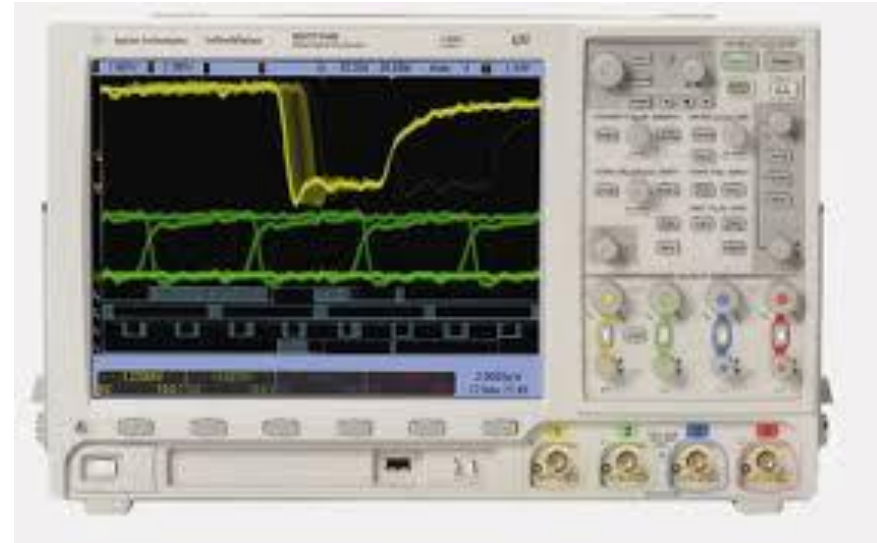
Human Machine Interface

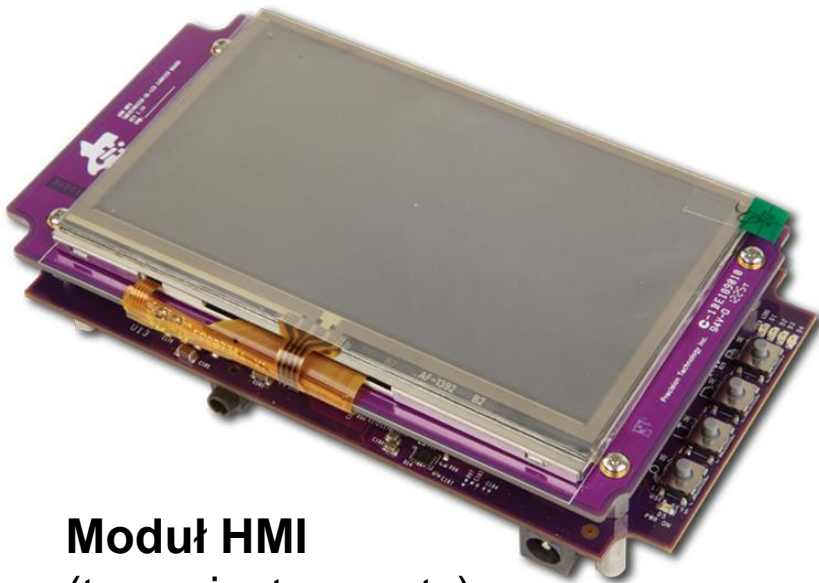


(texas instruments)

Oscyloskop „cyfrowy” (Digital Storage Oscilloscope)

połączenie FPGA i HMI





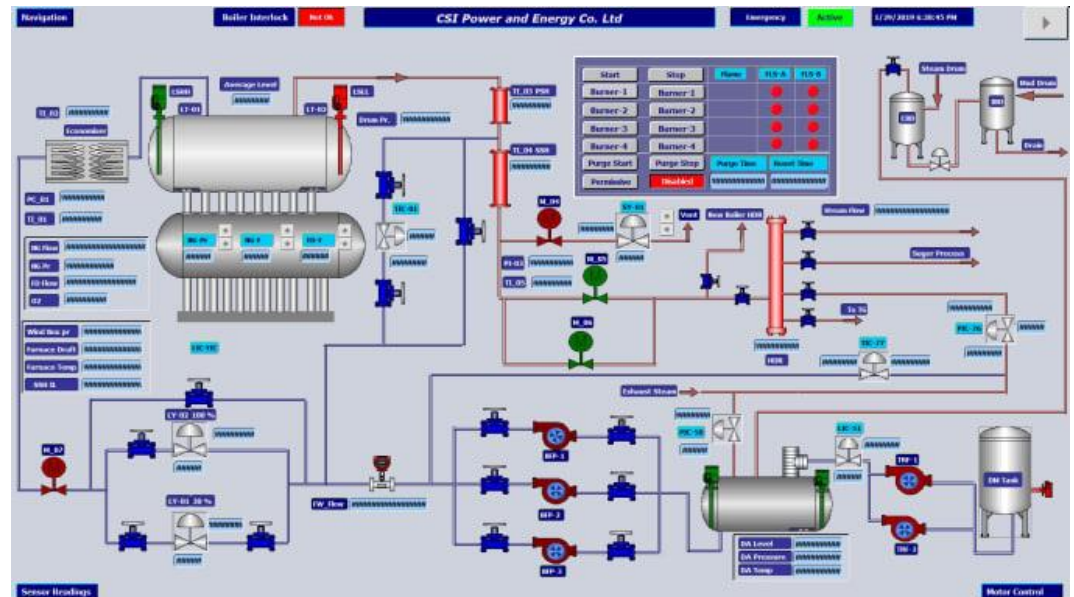
Moduł HMI
(texas instruments)



Automatyka przemysłowa

systemy **SCADA**

(Supervisory
Control and Data Acquisition)

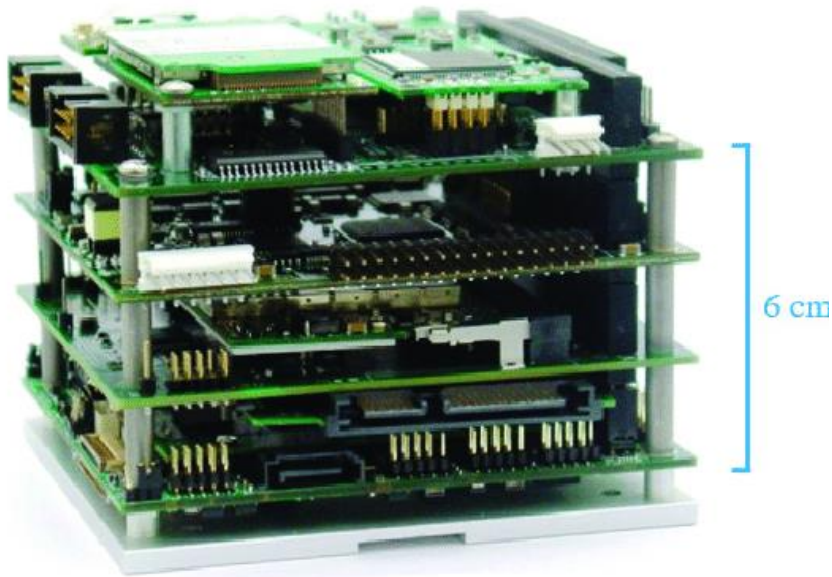


Komputery modułowe, przemysłowe

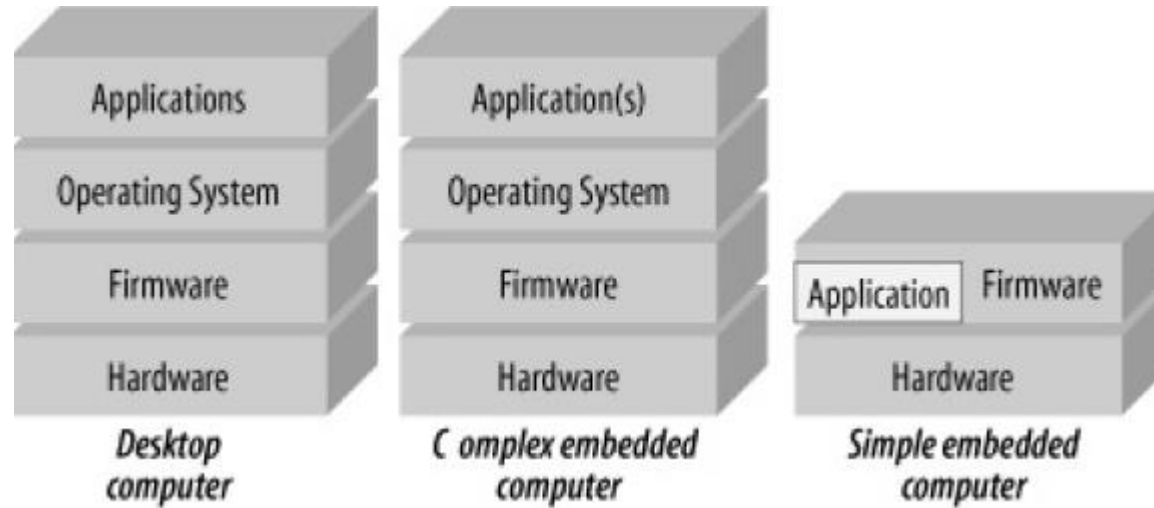
do pracy w specyficznych warunkach, do wbudowania, miniaturowe itp.

mikroprocesory „typowe”: x86, ARM, PowerPC itp.

Standardy np.: PC/104, compactPCI itp.



Systemy Wbudowane - software



„Wbudowane” systemy operacyjne:

Embedded Linux / OpenWrt (Wireless Router)

Windows Embedded / IoT / CE

FreeRTOS

...

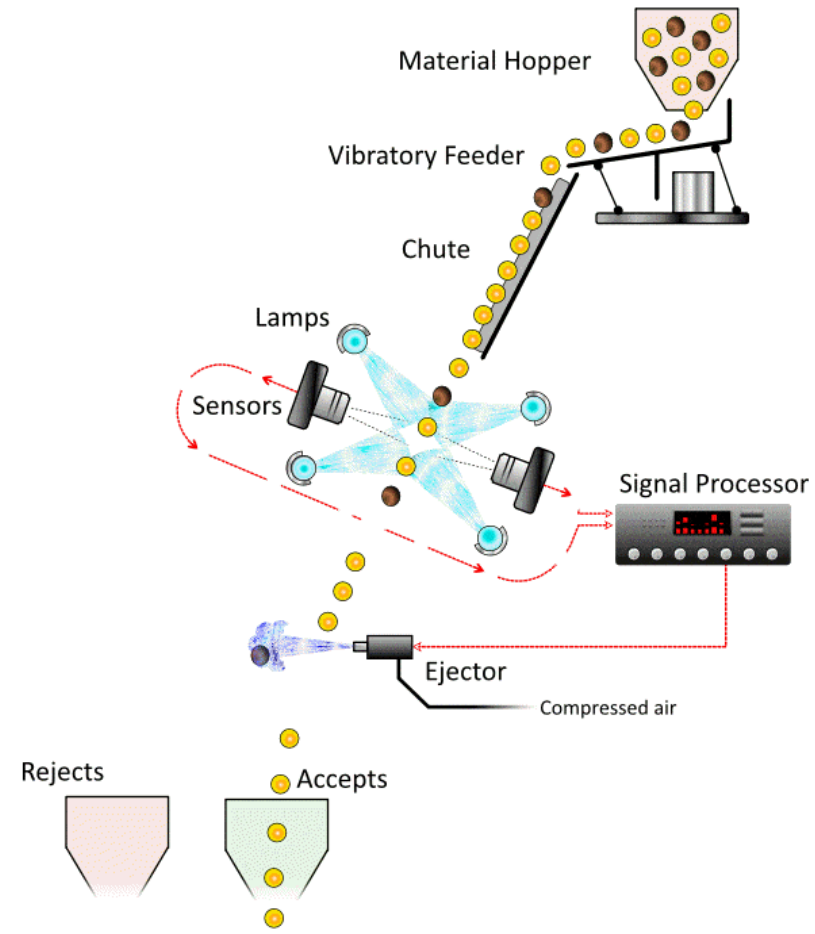
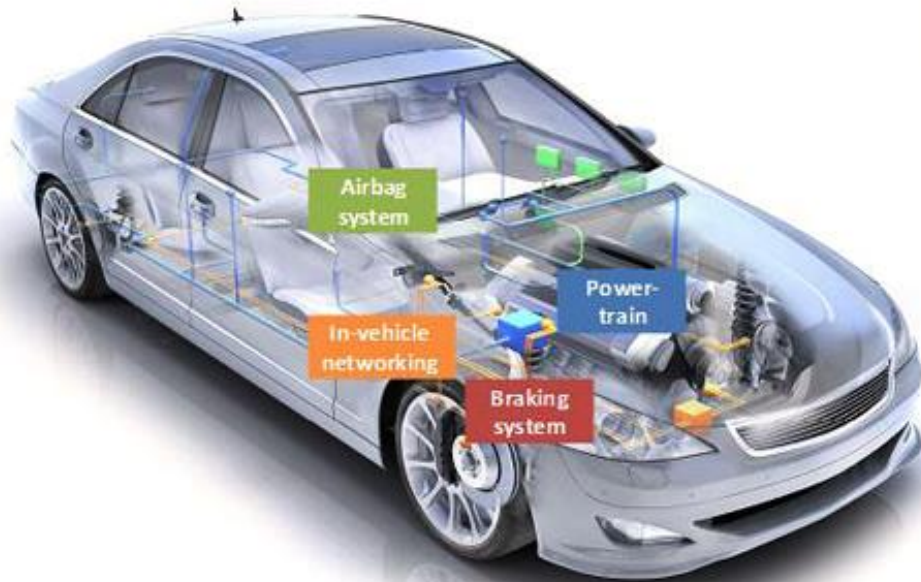
System Wbudowany jest przeważnie systemem czasu rzeczywistego

Przetwarzanie danych wejściowych i czas odpowiedzi - reakcja systemu nie może nastąpić później (czasami również nie wcześniej...) niż po ustalonym czasie.

Prawidłowe działanie zależy od jego logicznej poprawności, ale również od czasu odpowiedzi systemu na bodziec/sygnal wejściowy.

Miękkie sys. cz. rz. (Soft Real Time Systems)

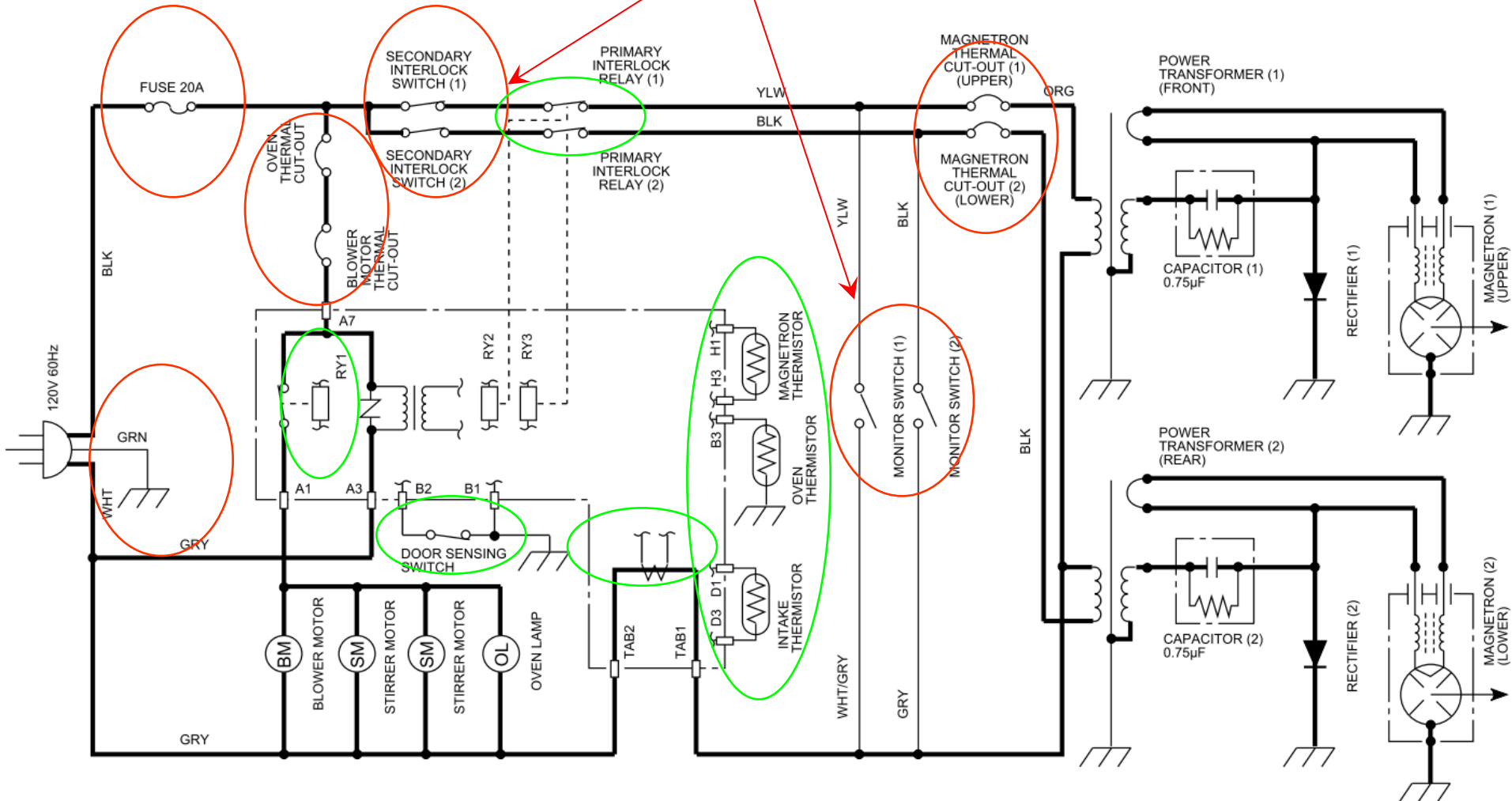
Twarde sys. cz. rz. (Hard Real Time Systems)



Bezpieczeństwo

Co można, a czego NIE POWIERZAĆ systemowi procesorowemu?

stan przełączników przy drzwiczkach zamkniętych

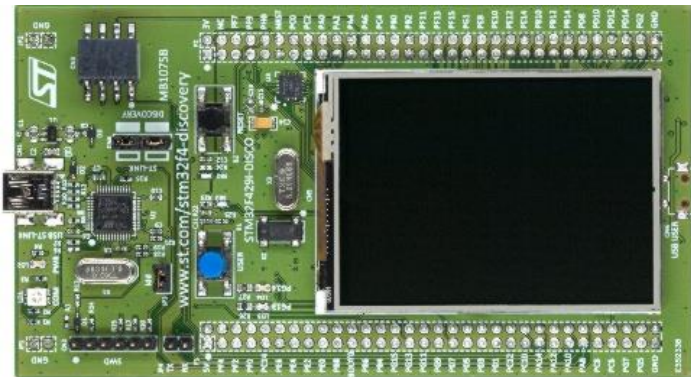


Wnioski

- Różnorodność obecnych zastosowań systemów wbudowanych, ich komplikacji i stawianych wymagań powoduje, że nie jest możliwy projekt jednej uniwersalnej platformy sprzętowej i programowej.
- W produkcji masowej oraz zastosowaniach profesjonalnych, optymalnym podejściem jest projekt i wdrożenie do produkcji dedykowanego, „skrojonego na miarę” sprzętu (wielkoseryjność -> redukcja kosztów, zast. prof. -> niezawodność, szybkość).
- W zastosowaniach hobbystycznych, małoseryjnych lub stawiających na uniwersalność i łatwość modyfikacji uzasadnione jest wykorzystanie gotowych platform sprzętowych i programowych (np. popularnych systemów operacyjnych).

Na laboratorium:

1. STM32F429I-Discovery
ARM Cortex M4



2. Arduino
AVR ATMEGA328

