Real-Time Systems

Exercise #1

Professor Jan Jonsson

Department of Computer Science and Engineering
Chalmers University of Technology

Laboratory assignment

Organization:

The laboratory sessions (one session/week in study weeks 4–8)

- To get guidance regarding the assignment problems, and also get help with debugging of software issues
- To demonstrate your solutions to each of the assignment problems, and have them approved

The project groups (3 students per group)

- The groups' weekly laboratory session takes place on:
  - RTS A xx: Mon @ 13:15-17:00
  - RTS B xx: Wed @ 08:00-11:45
  - RTS C xx: Wed @ 15:15-19:00*

  (* study weeks 5 & 7: Wed 17:15-21:00)

Location:

Jupiter building,
campus Lindholmen

Laboratory assignment

Organization (cont'd):

The loan equipment

- You will be offered loan equipment by the department, and work on the assignment at home on your own computer.
- The loan equipment for each project group consists of:
  - one MD407 processor card
  - one USB cable + one CAN cable
- You pick up the loan equipment during study week 3, and return the loan equipment during exam week.
- The equipment is picked up in the lab room at Lindholmen campus during your scheduled lab time slot. To return the equipment you book an appointment with the examiner.

Laboratory assignment

Basic prerequisites for approval:

Respecting the Rules of Conduct (see separate document)

- Contribute to your group: (be present, be active)
- Respect the laboratory sessions: (read guidelines / lab-PM)
- Respect the deadlines: (submit assignments in time)
- Refrain from cheating:
  Do not copy other people's code!
  Do not share your code with other groups!

Laboratory performance of satisfactory quality (≥ grade 3)

Laboratory assignment

Grading of the ‘Laboratory’ course element:

The grade (U, 3, 4, 5) will reflect your practical skills at the laboratory sessions.

The grade is determined by the following:

- The quality of laboratory performance
  - sub-score is awarded* based on a set of four criteria
  - sub-score sets the final grade

(* See corresponding modules in Canvas for details)

Laboratory assignment

Laboratory performance sub-score criteria:

Implementation

- How many of the coding challenges in Part 2 that you can successfully implement and demonstrate.

Design

- How well you know the design and behavior of your code.

Debugging

- How well you identify, and solve, problems with your code.

Paradigm

- How well you understand, and make use of, the Concurrent, Object-oriented, Reactive, Timing-aware programming method.

Laboratory assignment – Part 0

Getting started: [compulsory]

- Compile the template code using the cross compiler
- Upload the machine code to the target computer

Interacting with the target computer: [compulsory]

- Take input from the workstation's keyboard
- Generate output to the workstation's console window

Preparatory work for Part 1 and Part 2: [compulsory]

- Pre-compute periods for all tones that will be played
- Prepare data structures to allow a melody to be transposed to different keys

Laboratory assignment – Part 1

Tone generator: [compulsory]

- Generate a 1 kHz tone (square wave signal) and output it to the audio jack on the target computer

Background load: [compulsory]

- Add a background task with a scalable load
- Experiment: disturb tone generator by increasing the load
- Repeat the experiment with deadline scheduling enabled

Worst-case execution times: [compulsory]

- Measure the execution times of the program code in the background load task and the tone generator task

Laboratory assignment – Part 2

Brother John music player: [compulsory]

- Capable to play tones in a 12-tone scale in different keys
- Capable to play the melody “Brother John”, and be able to change key and/or tempo while playing

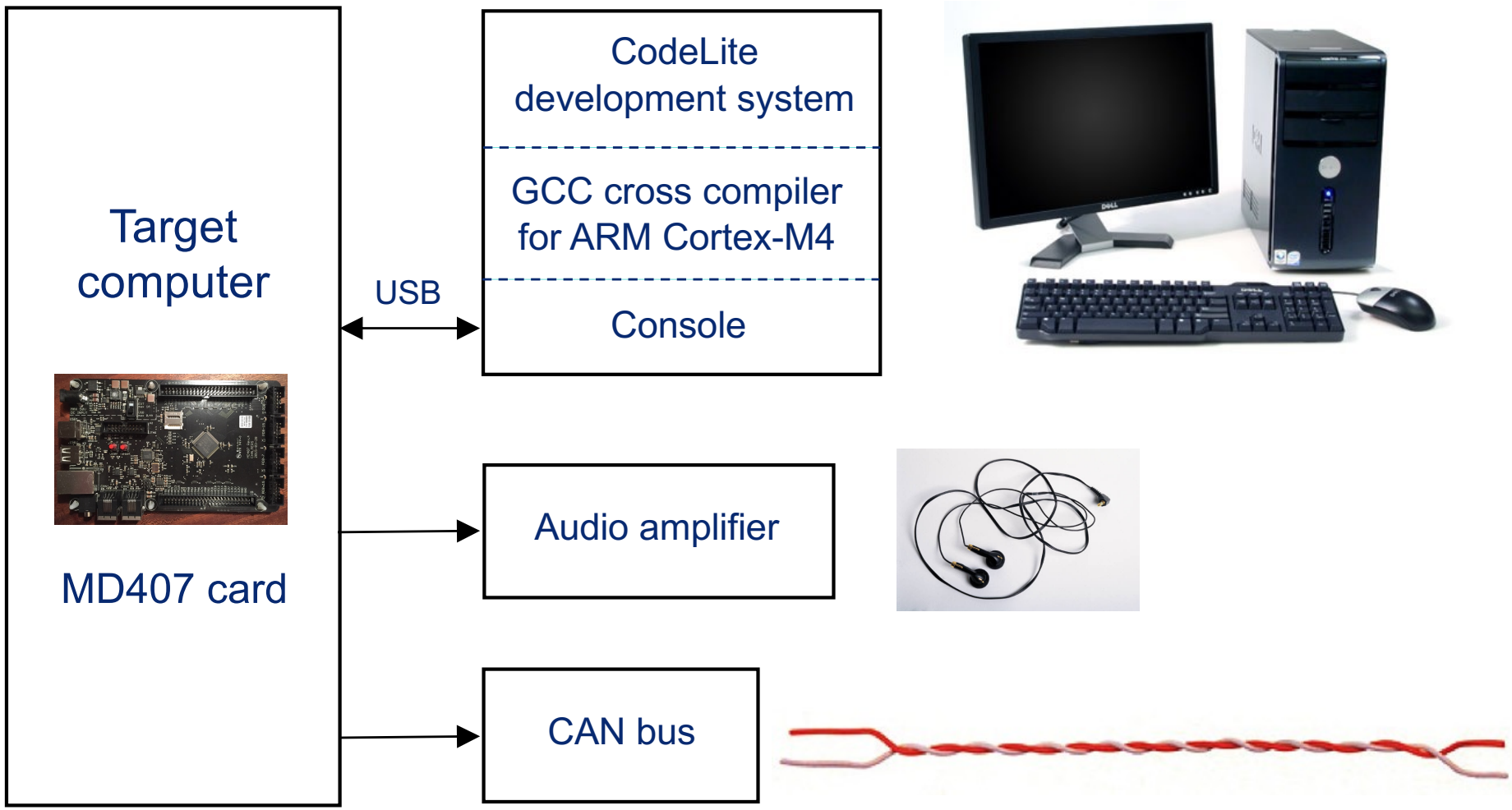
Loopback CAN: [required for grade 4 or higher]

- Capable to control the music player via the CAN bus
- Capable to run in two different modes: conductor / musician

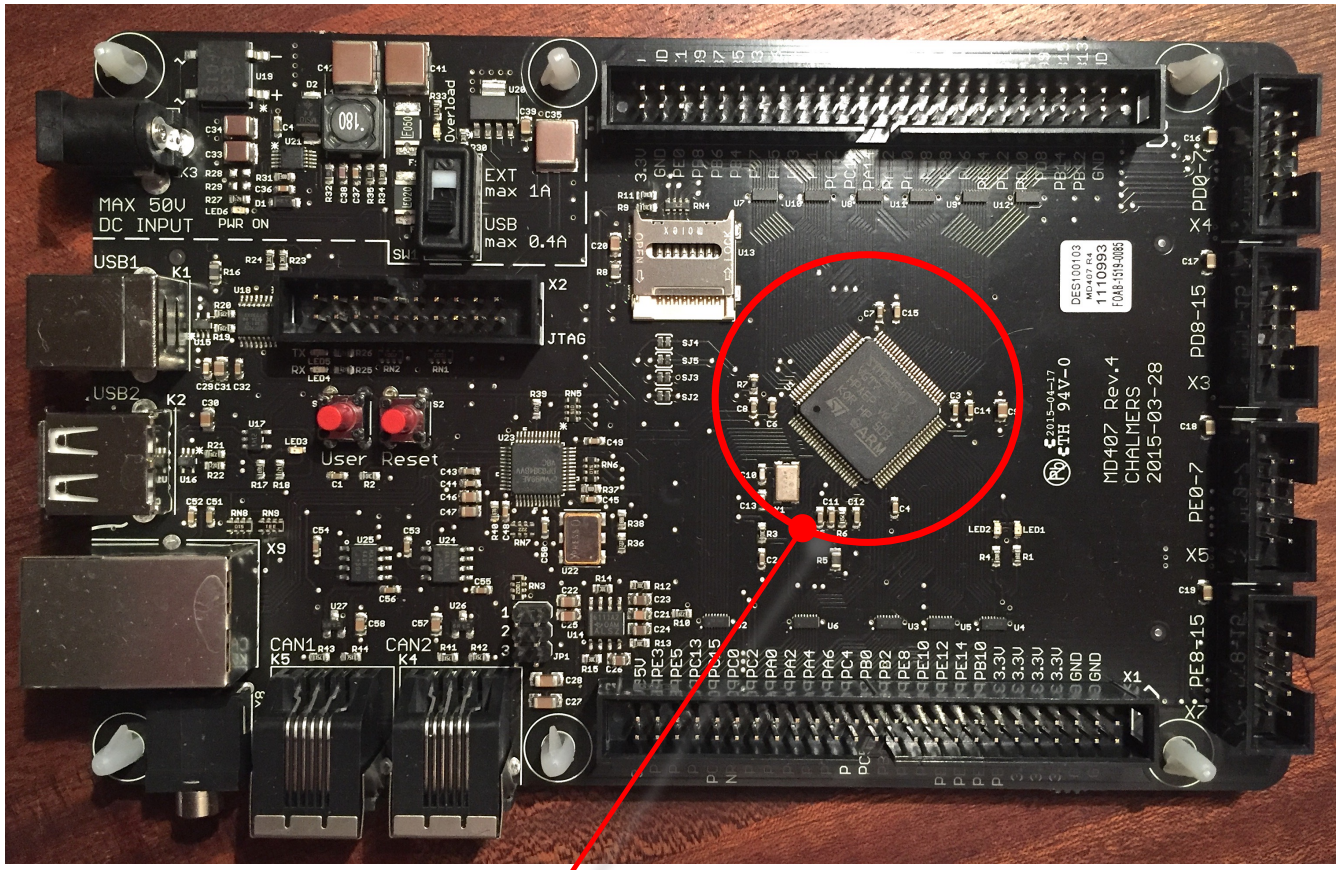
Tap/Blink tempo with button/LED : [required for grade 5]

- Extend the music player with extra interactive functionality
- Utilize red “user” button and green LED on processor card

Laboratory assignment – Setup

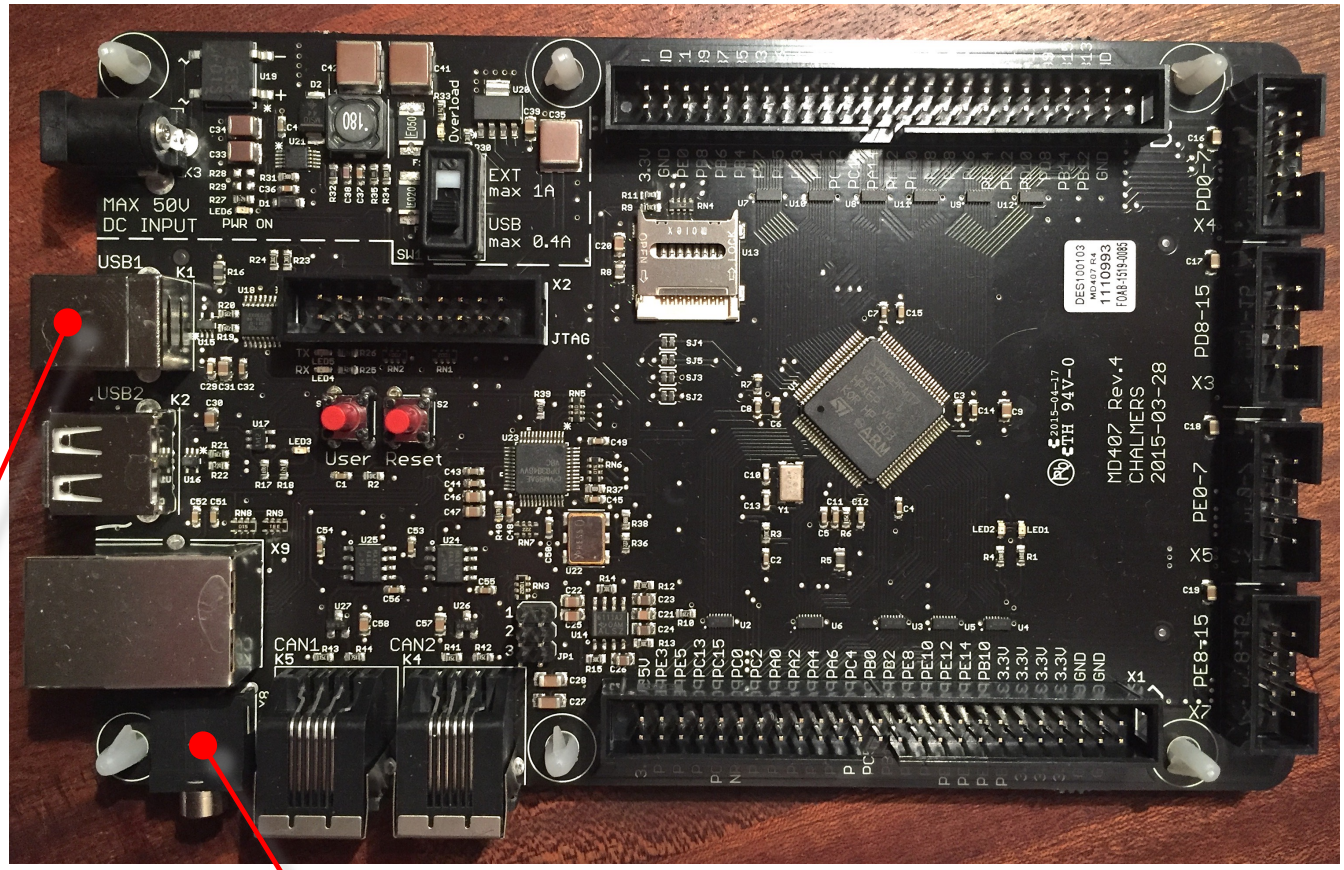


The MD407 computer card



STMicroelectronics' STM32F407 microcontroller /w ARM Cortex-M4 core

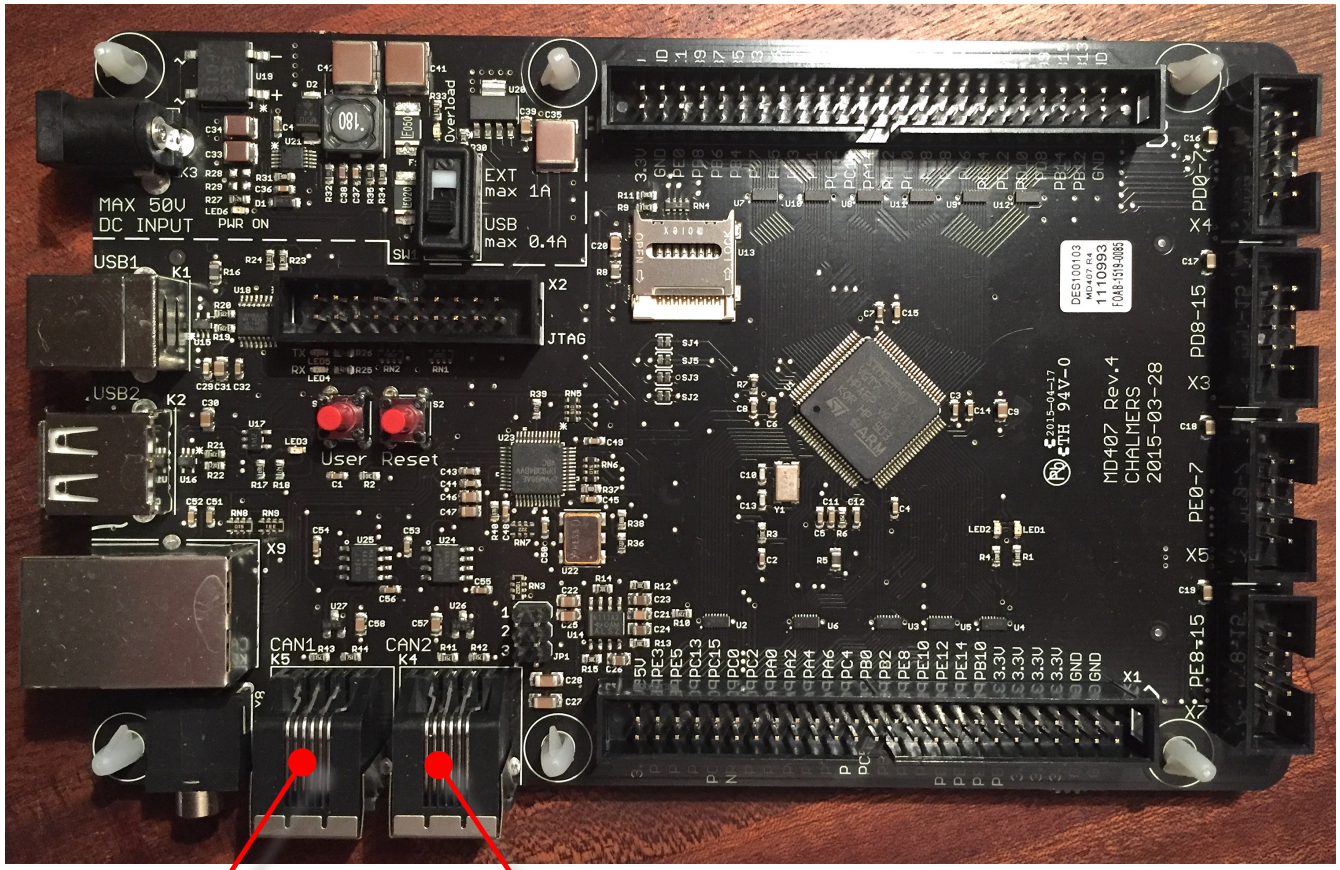
The MD407 computer card



USB debug port

Audio output jack

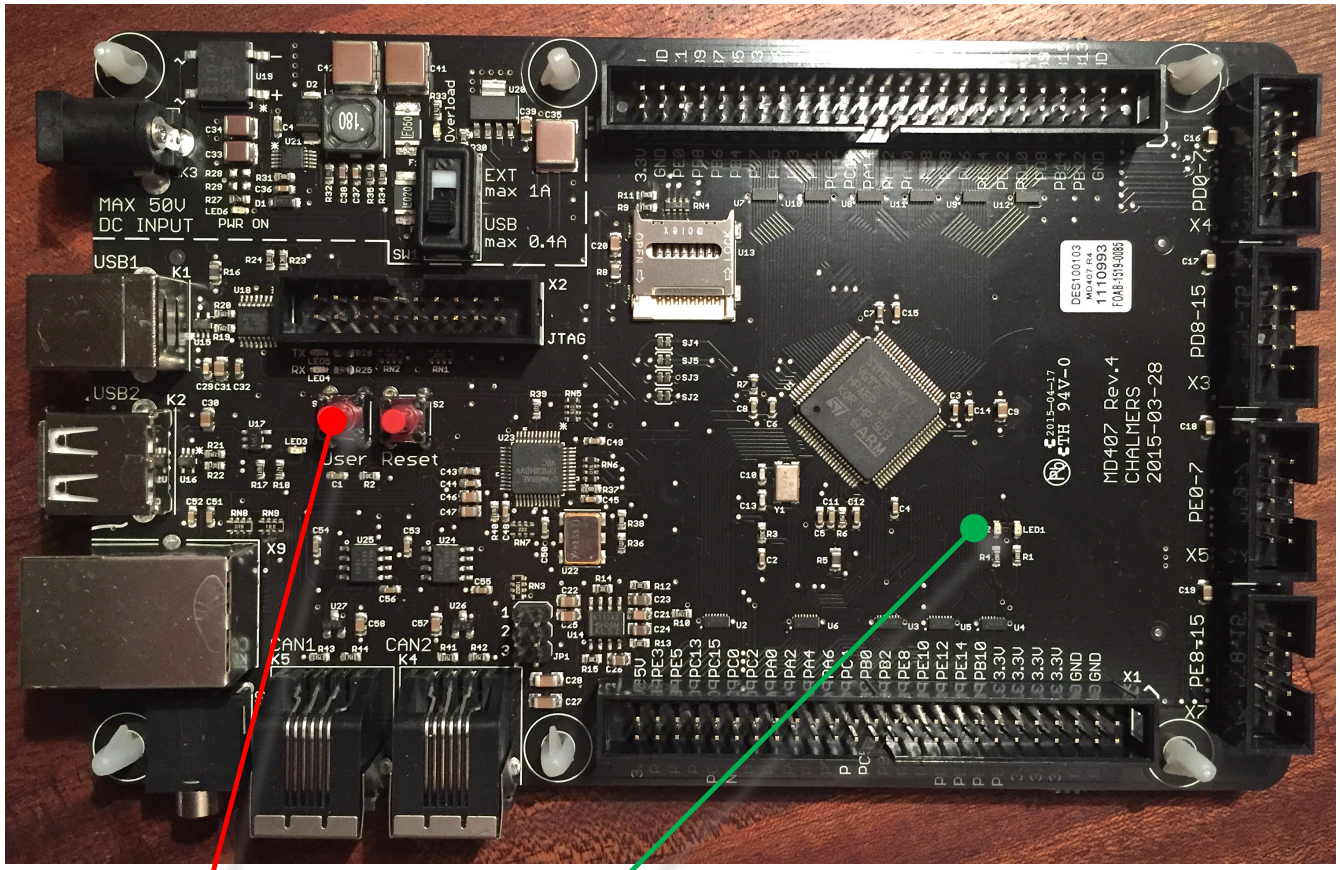
The MD407 computer card



Main CAN bus

Loopback CAN bus

The MD407 computer card



Red "User" button

Green LED

The MD407 computer card

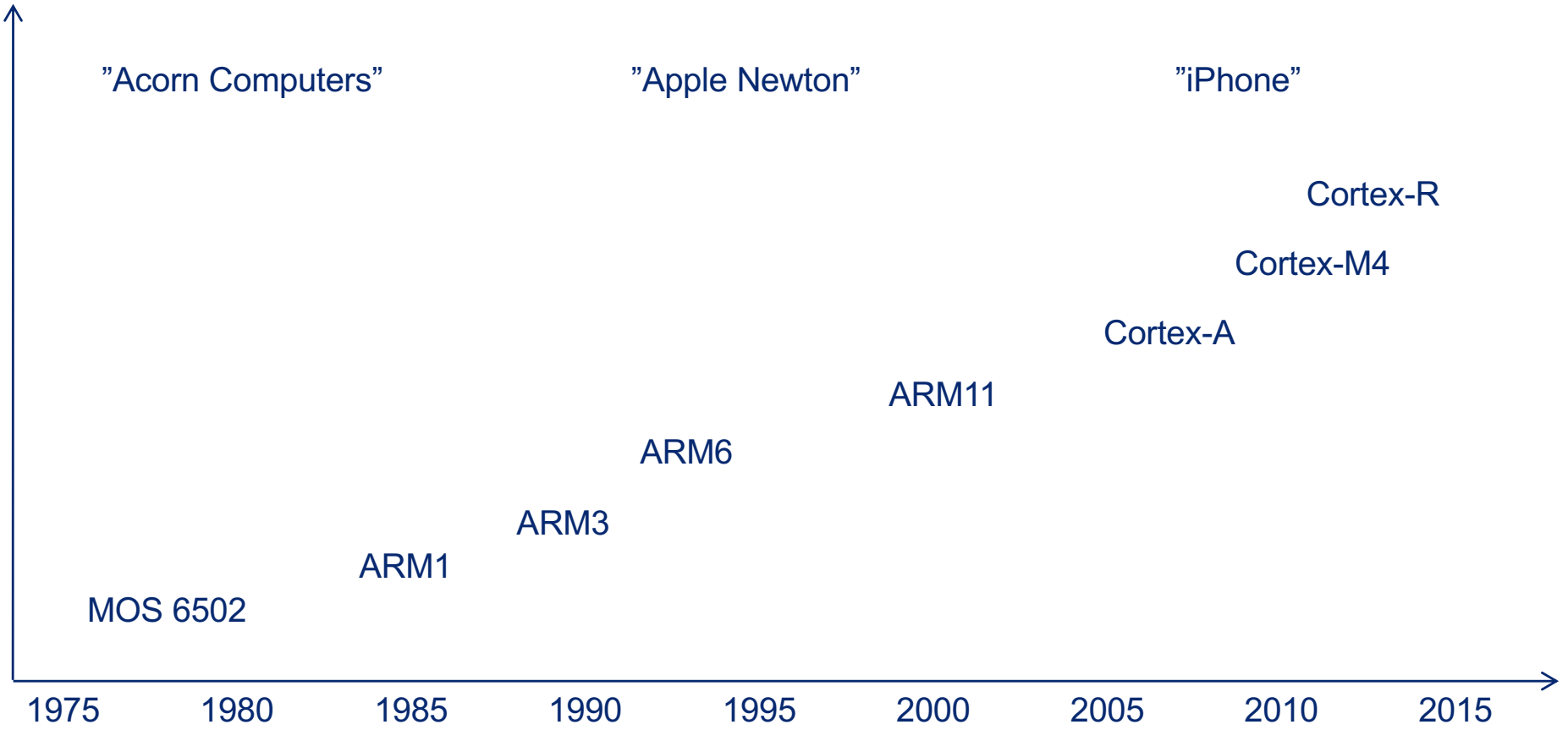
The STM32F407 microcontroller:

Based on the ARM Cortex-M4 processor core

- 168 MHz processor clock
- 32-bit registers (data and address)
- 16-bit instructions (Thumb)
- 1 MB of Flash memory (for resident monitor/debugger)
- 128 kB of RAM (for user programs)
- On-chip floating-point unit
- On-chip CAN modules, serial communications interfaces, parallel ports, digital-to-analog converters, high-resolution timers, ...

The MD407 computer card

The ARM processor family tree:



STM32F407 block diagram:

CPU – Cortex-M4 core

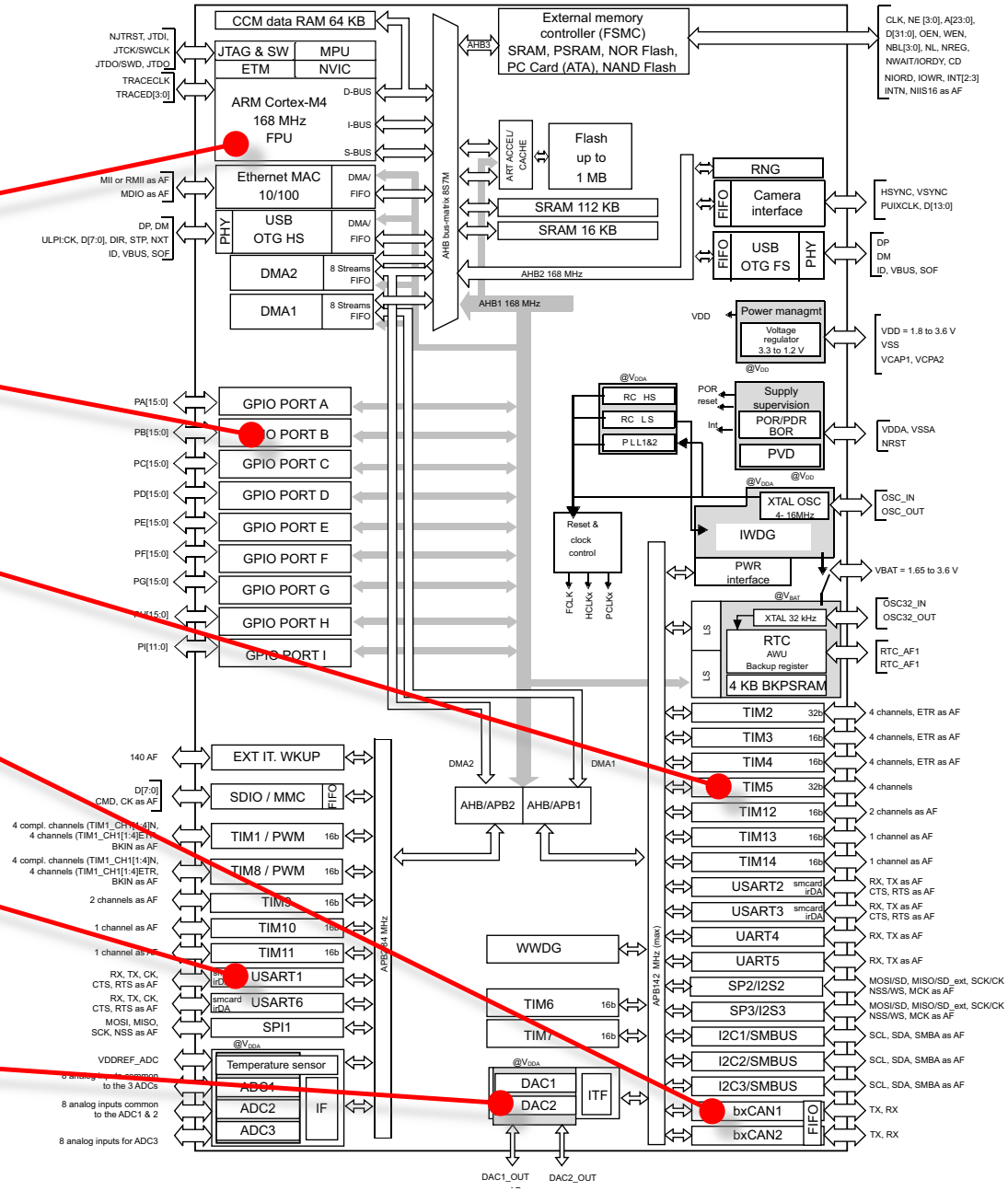
Port B – connects to red “User” button and green LED

TIM5 – used by TinyTimber for high-resolution clock

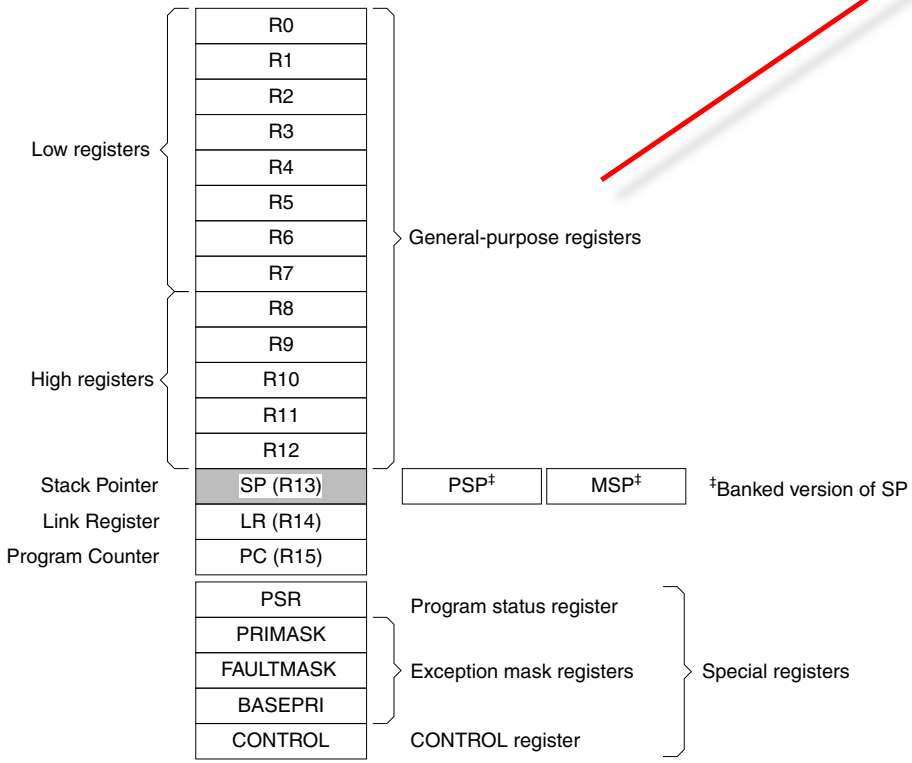
bxCAN1 – available for your connection to other cards

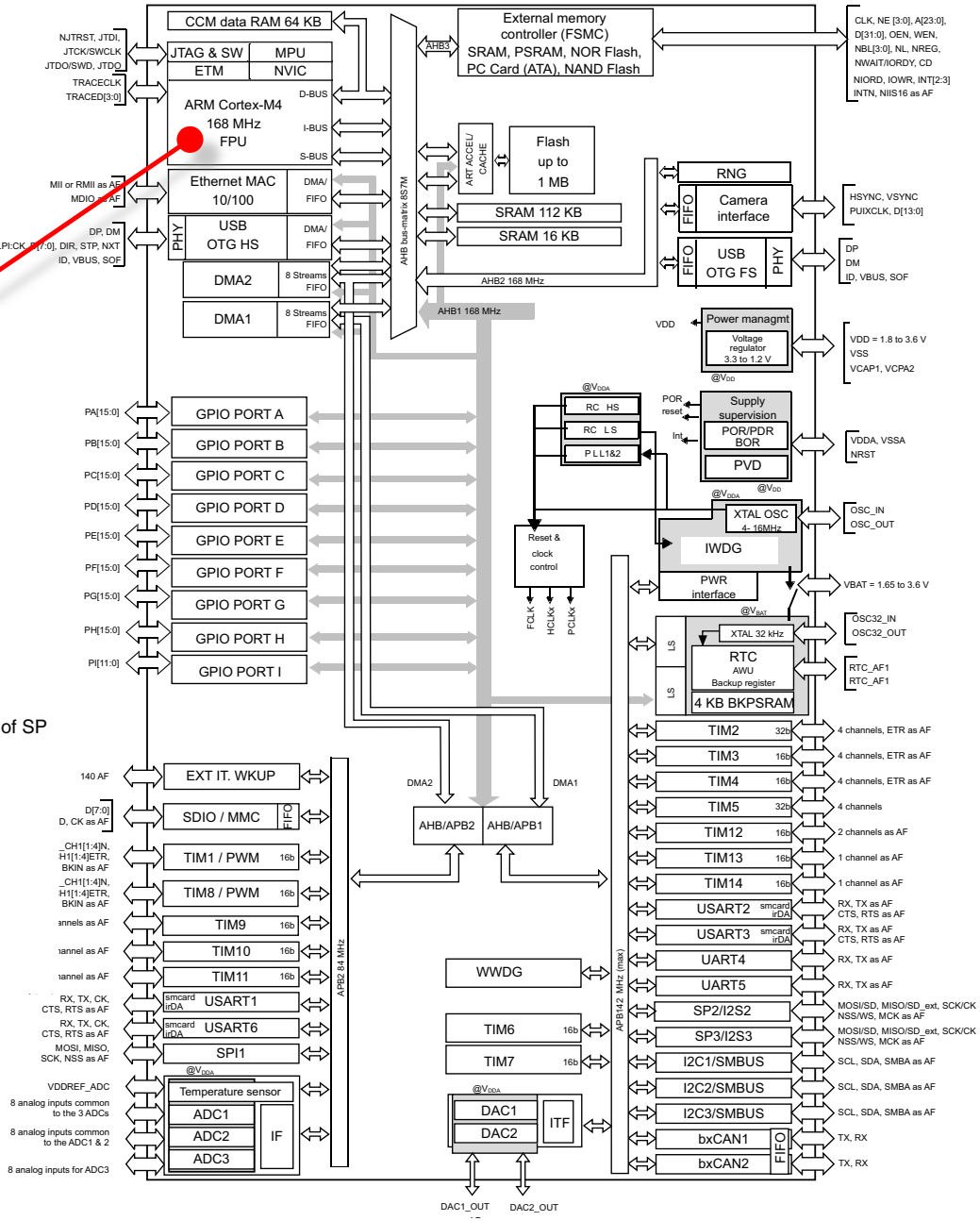
USART1– connects to console via USB debug port

DAC2 – available for your sound generation

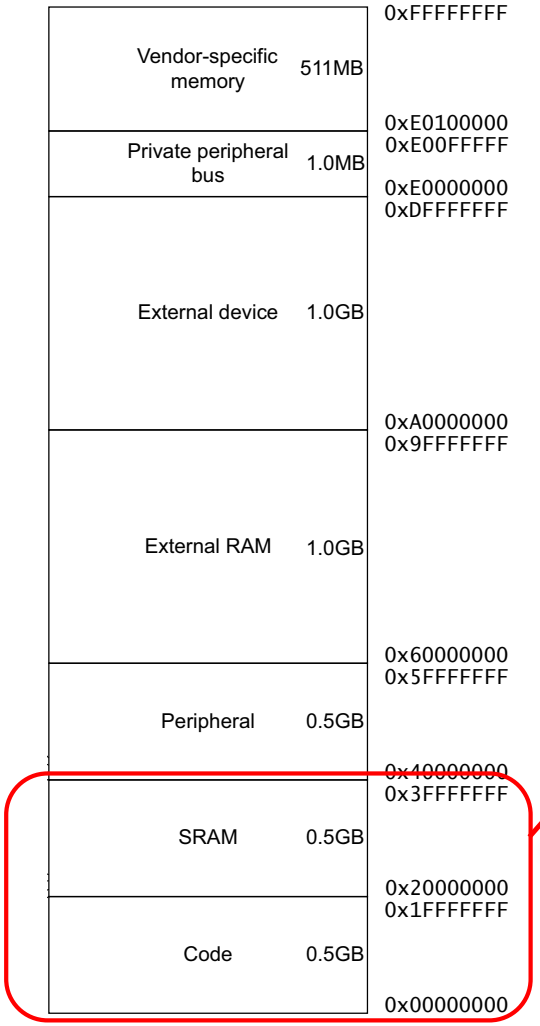


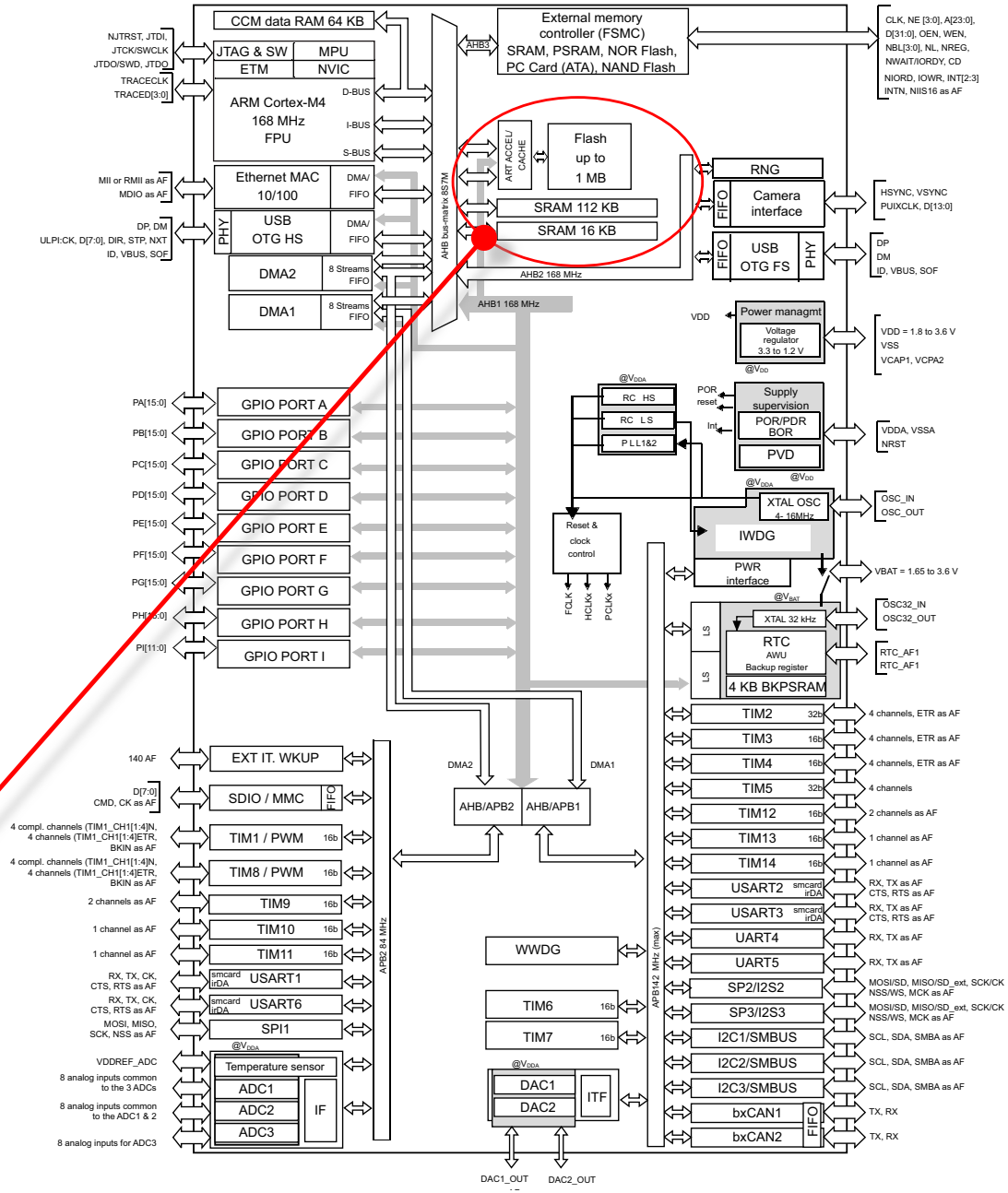
STM32F407 processor registers:





STM32F407 address space:





STM32F407 address space: