# Real-Time Systems

## Exercise #1

Professor Jan Jonsson

Department of Computer Science and Engineering
Chalmers University of Technology

# Laboratory assignment

## Organization:

### The schedule

- The laboratory sessions run weekly, starting in study week 2
- Each project group is guaranteed one session per week
- Deadline to register for a project group is Monday @ 09:00

### The parts

- The assignment consists of Part 0, Part 1 and Part 2
- The parts normally take a total of 6 sessions to complete
- Part 0 typically takes 1 session to complete
- Part 1 typically takes 1−2 sessions to complete
- Part 2 typically takes 3−4 sessions to complete

# Laboratory assignment

## Basic prerequisites for approval:

### Respecting the Rules of Conduct (link on 'Assignment' page)

- Show respect for others in the laboratory room
- Refrain from cheating

### Sufficient lab session attendance

- Since the assignment involves significant interaction with laboratory assistants as well as other groups each student is expected to attend at least three laboratory sessions, the first attendance being no later than study week 3.

### Project report of sufficient quality

- Should contain description of solutions to selected design challenges in the lab-PM (see 'Assignment' page for details)

# Laboratory assignment

The grade will be based on your performance in:

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 can make use of, the reactive, concurrent and timing-aware programming paradigm.

# Laboratory assignment – Part 0

## Getting started:

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

## Interacting with the target computer:

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

## Preparatory work for Part 1 and Part 2:

- 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:

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

## Background load:

- 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:

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

# Laboratory assignment – Part 2

## Single instrument: [each group individually]

- 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

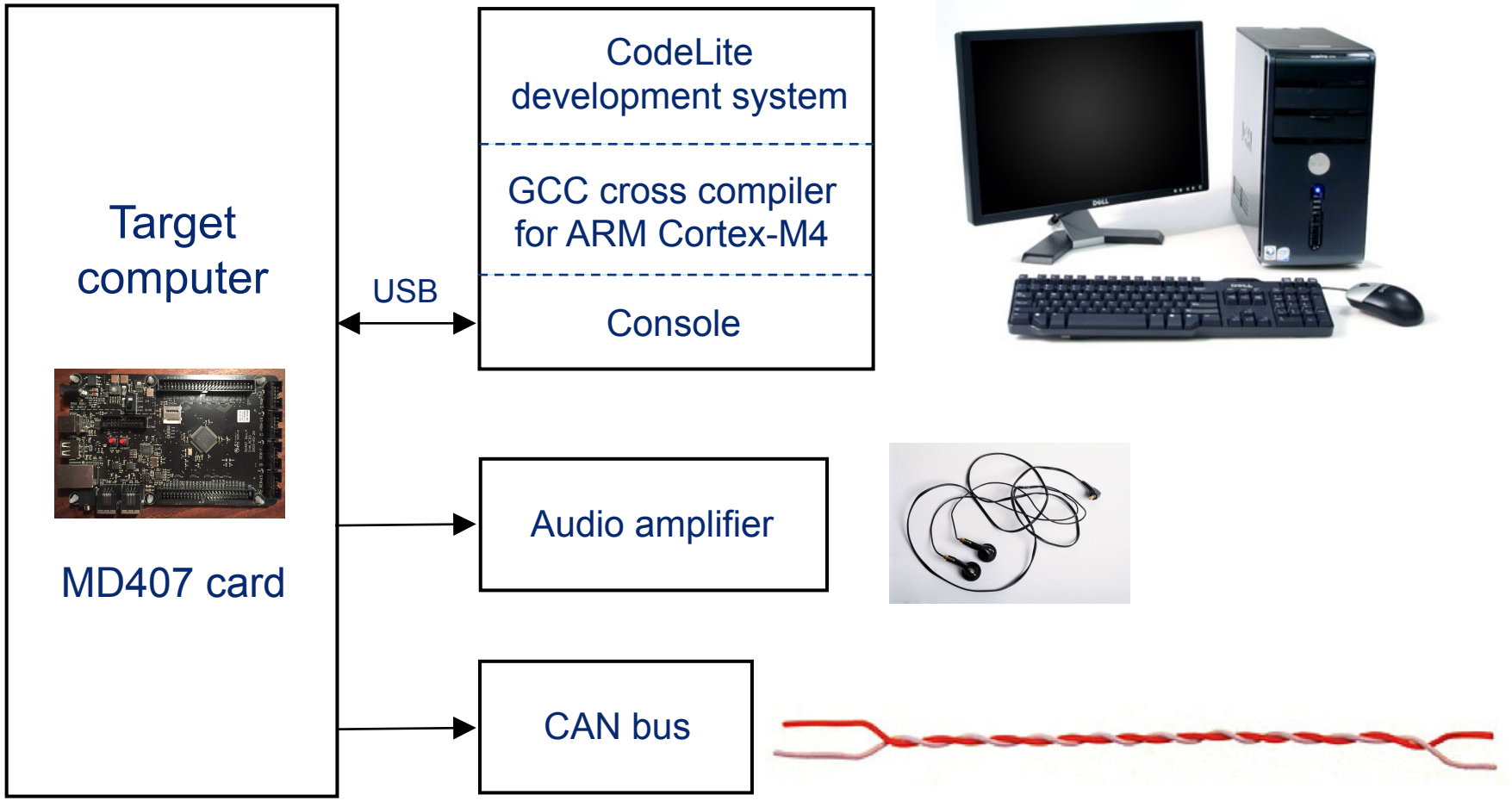
## Basic orchestra: [collectively, with two or more computers]

- Capable to play “Brother John” in chorus form (unison), with one target computer being the orchestra leader (conductor)
- Conductor should be able to set initial key and tempo

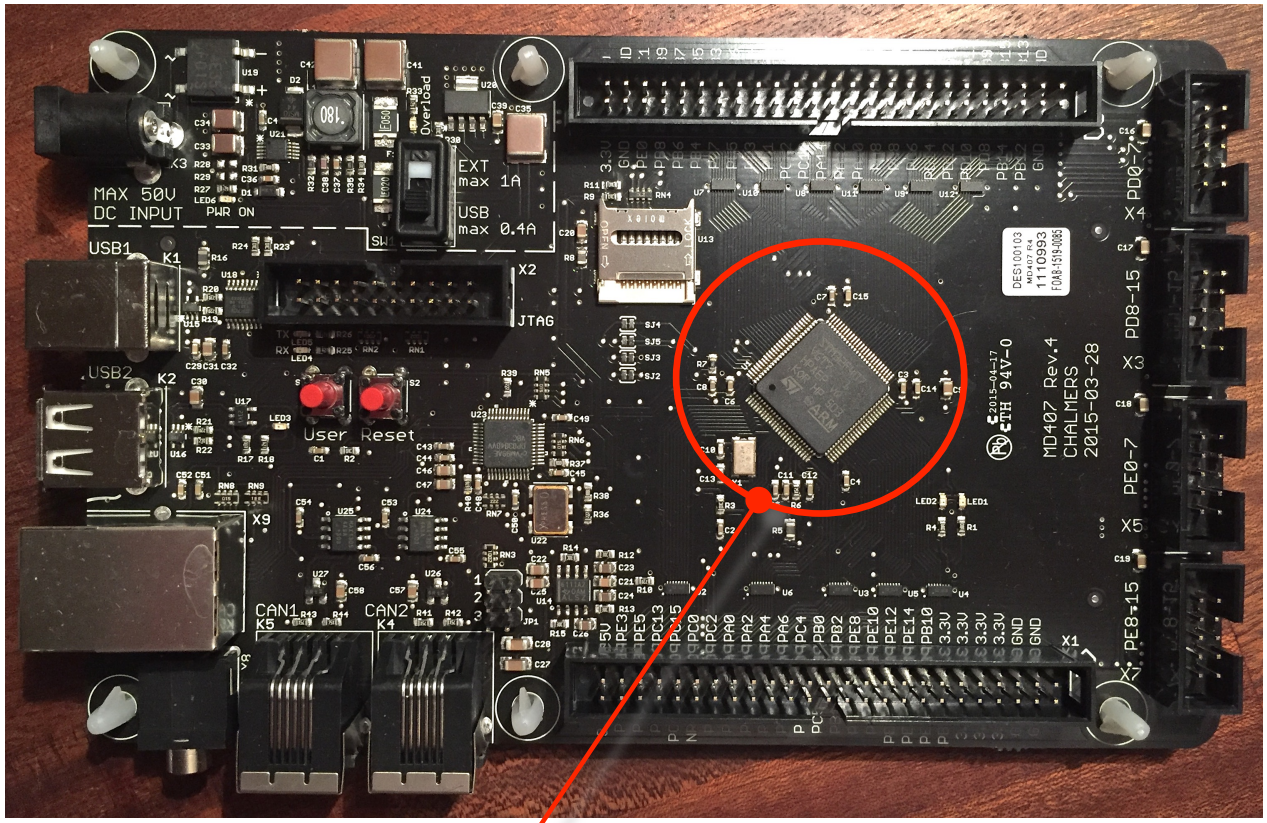
## Advanced orchestra: [collectively]

- Play in canon form, with conductor role as above
- Conductor should be able to change tempo dynamically

# Laboratory assignment – Setup

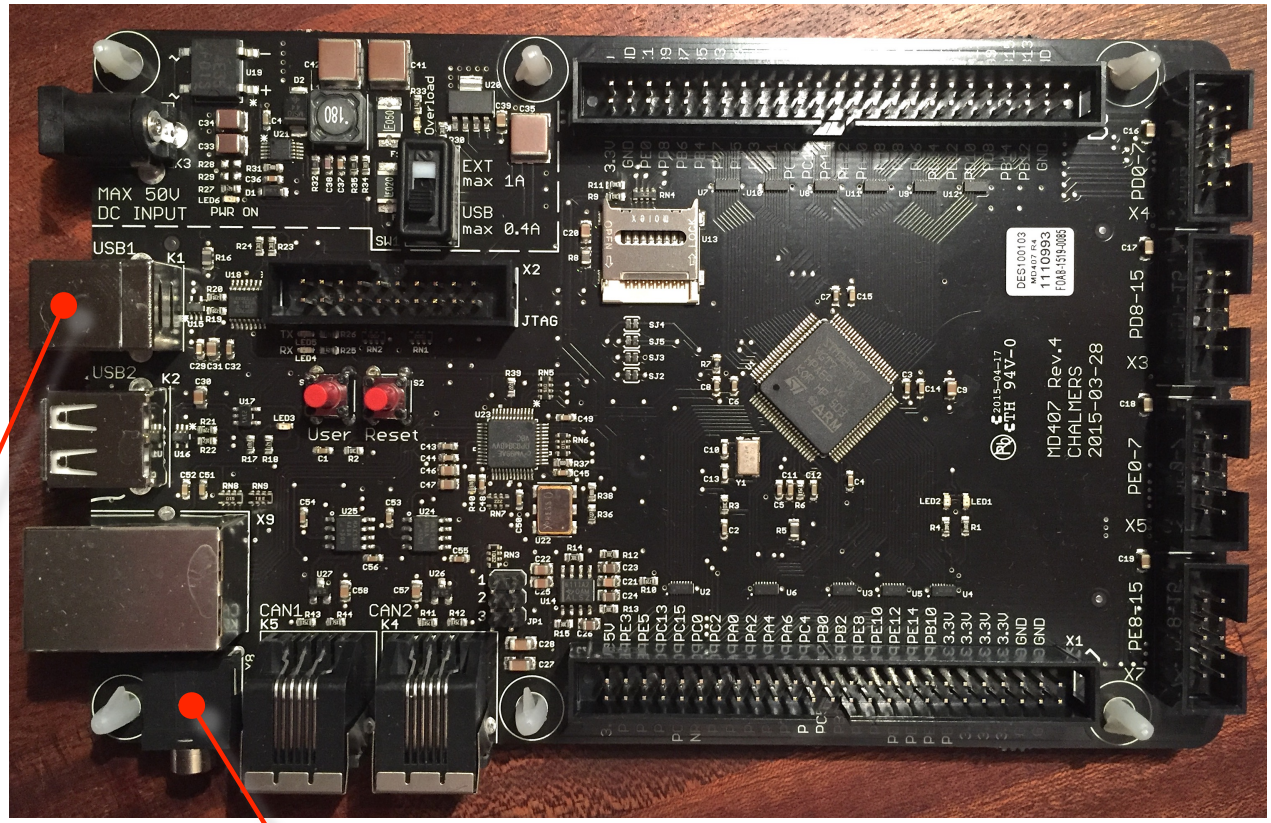


# The MD407 computer card




STMicroelectronics' STM32F407 microcontroller /w ARM Cortex-M4 core

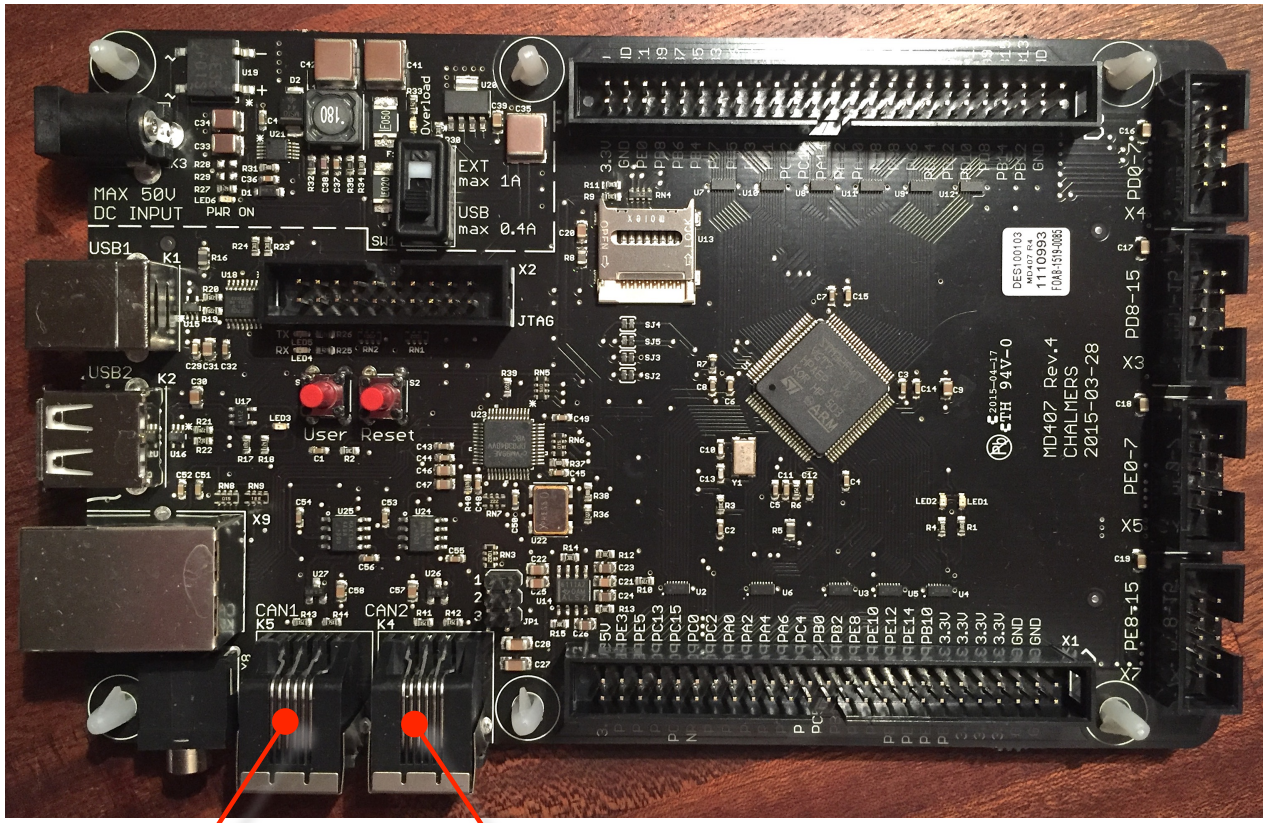
# The MD407 computer card




USB debug port

Audio output jack

# The MD407 computer card




Main CAN bus

Loopback CAN bus

# The MD407 computer card

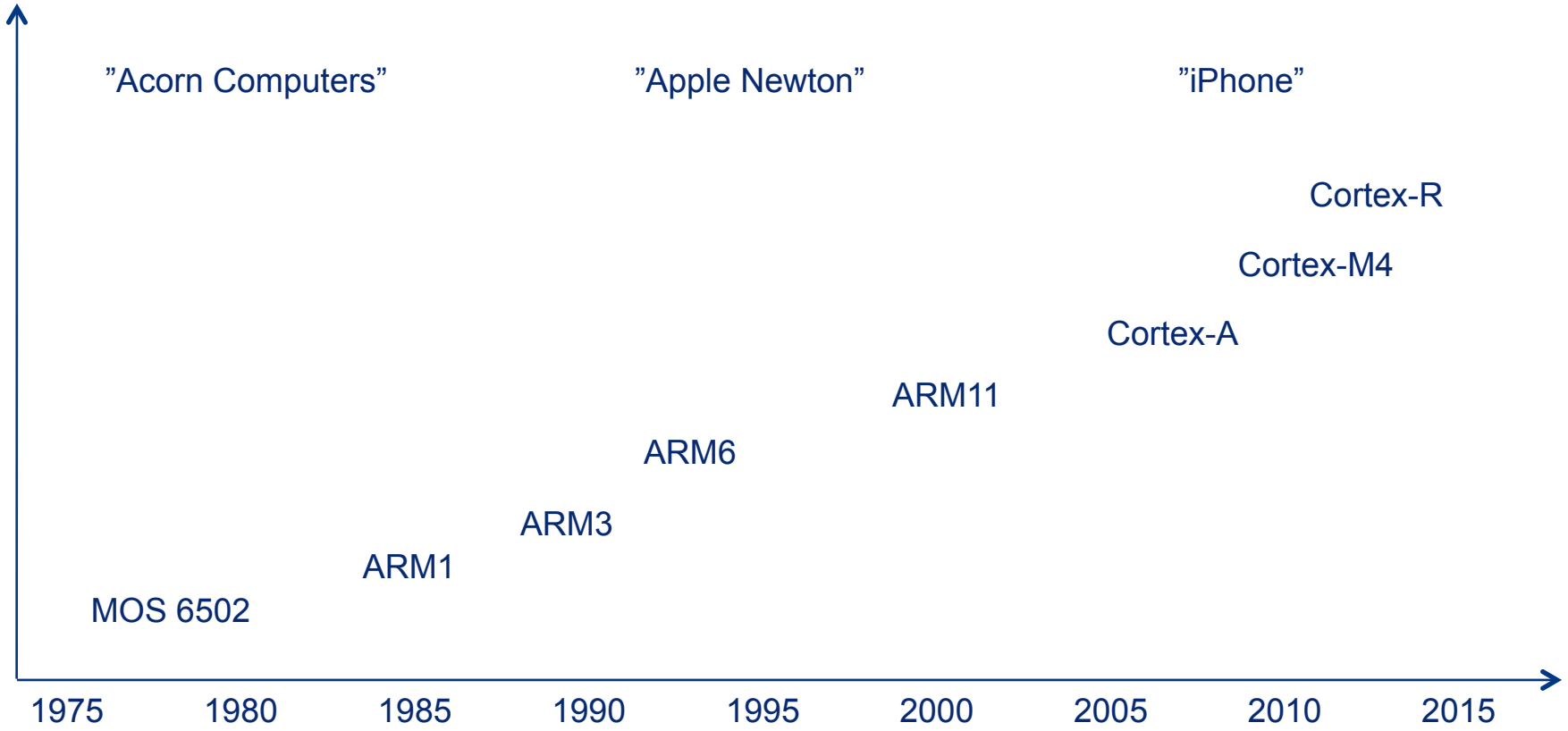
## The STM32F407 microcontroller:

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:

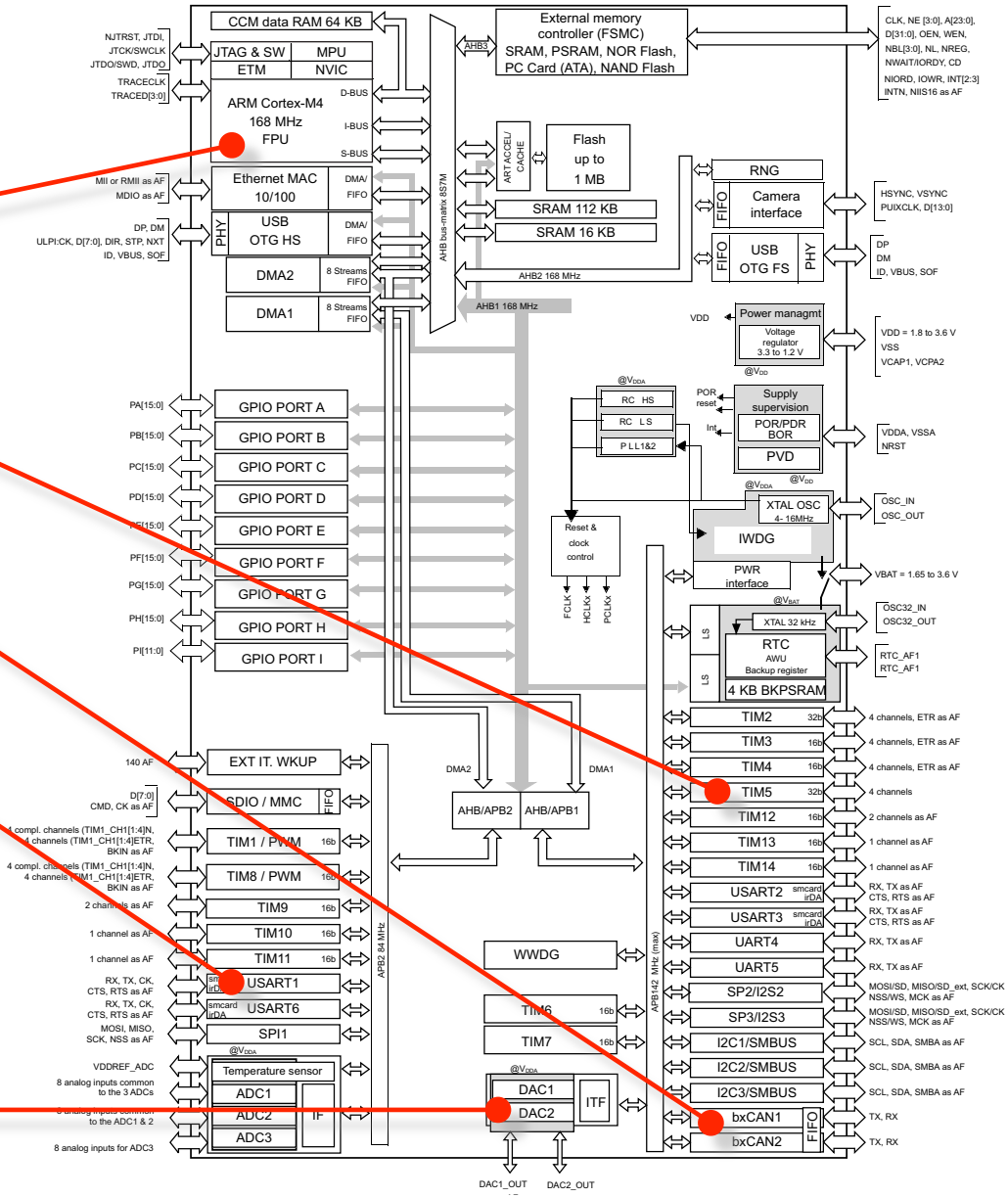
CPU – Cortex-M4 core

TIM5 – used by TinyTimber for high-resolution clock

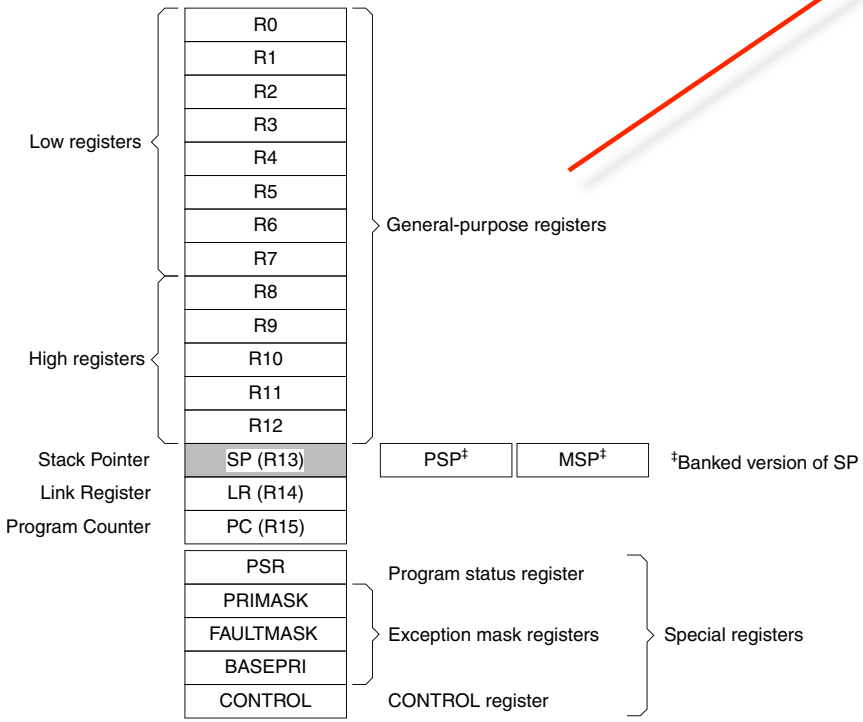
bxCAN1 – available for your connection to other cards

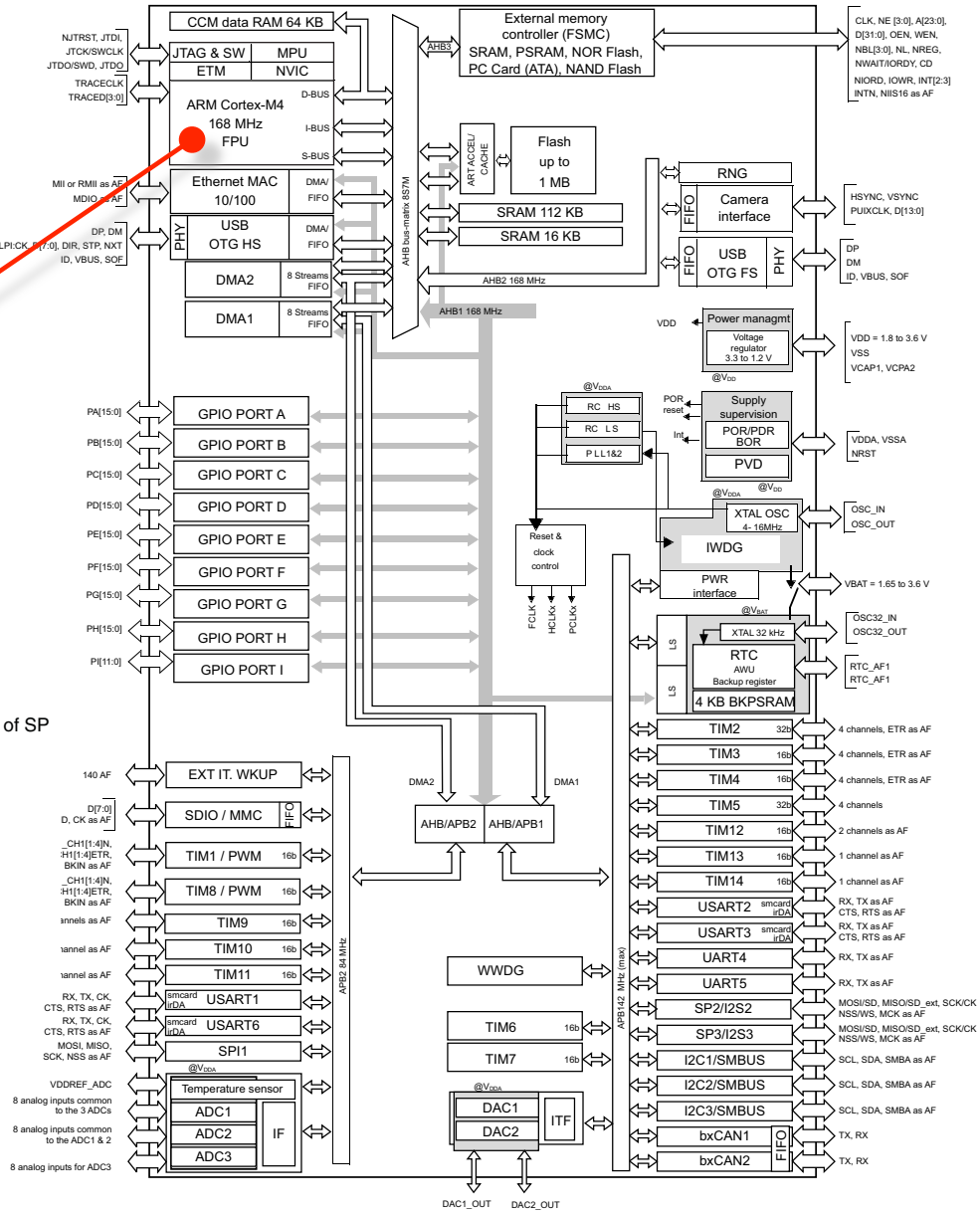
USART1– connects to console via USB debug port

DAC2 – available for your sound generation

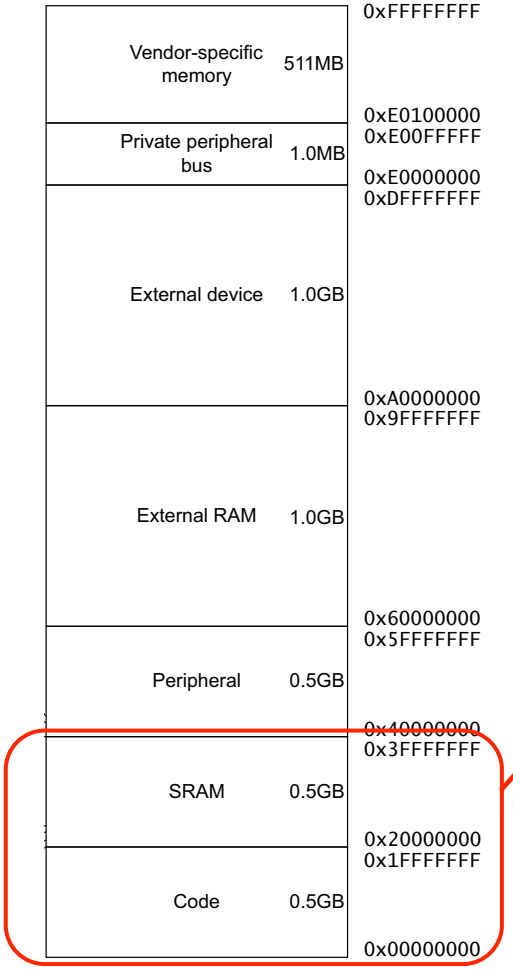


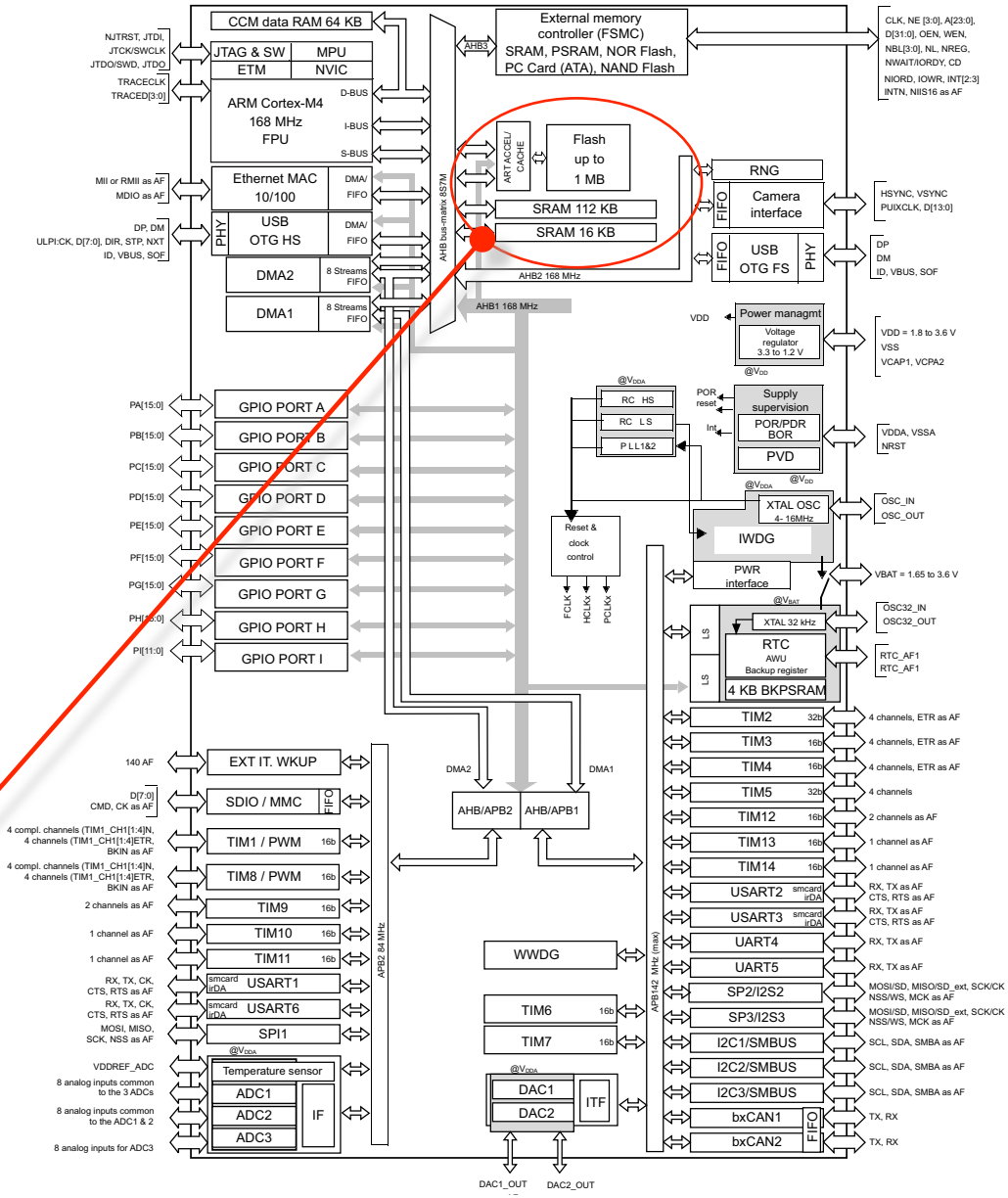
# STM32F407 processor registers:



# STM32F407 address space:



# STM32F407 address space: