PV198 - I²C

One-chip Controllers

Daniel Dlhopolček, Marek Vrbka, Jan Koniarik, Oldřich Pecák, Tomáš Rohlínek, Ján Labuda, Jan Horáček, Matúš Škvarla

Faculty of Informatics, Masaryk University

7/2023

Intro

- Switch the branch to *Week_07*!
- Discussion of HW6

NXP CUP

- International competition with autonomous cars
- Working on a group project
- Diverse set of tasks, for example:
  - Parse data from sensors
  - Constructing platform holding sensors
  - Design controlling algorithm
- Last year finals
- Contact:
  - Email: *jan.labuda@mail.muni.cz*
  - Discord: *Ján Labuda / Northeus*

Embedded communication buses

- SPI – Serial Peripheral Interface
- **I²C – Inter-Integrated Circuit**
- UART – Universal asynchronous receiver / transmitter
- CAN – Controller Area Network
- 1-Wire
- RS-485
- RS-232

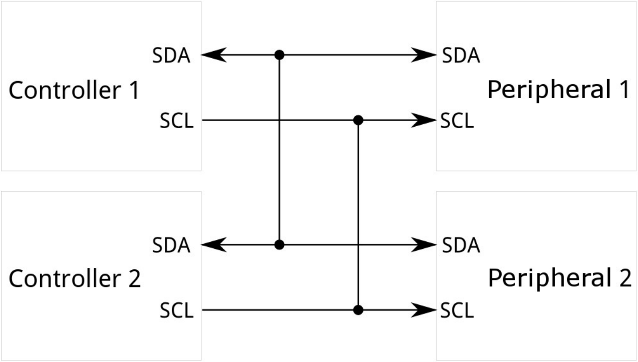
I^2C

- "eye-squared-C"
- I^2C = Inter-Integrated Circuit
- Invented in 1982 by Philips Semiconductors (now NXP Semiconductors)
- Applications: intra-board communication
  - Peripherals
  - Sensors

I^{2}C Principle

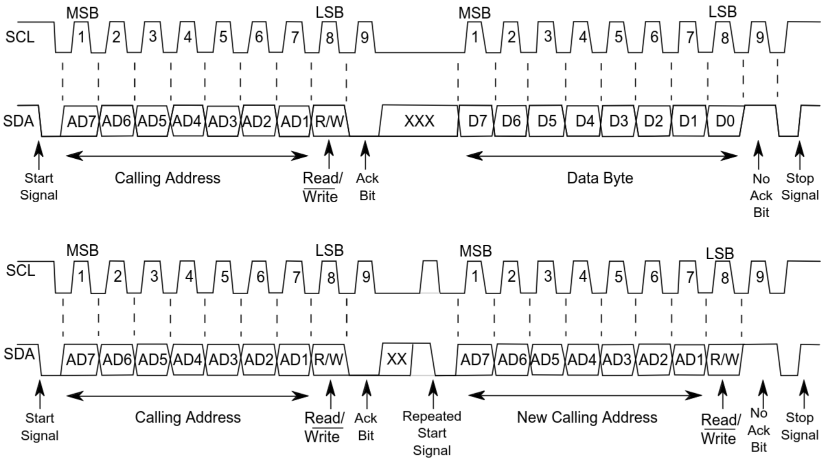
- 2 wires
  1. SCL – serial clock
  2. SDA – serial data
- Terminology: controller (= master), target (= slave)
- Multi-controller & multi-target
- 100 kbit/s – 5 Mbit/s
- 7-bit addressing / 10-bit addressing
- Synchronous
- Half-duplex

I^2C Schematic



Source: https://learn.sparkfun.com/tutorials/i2c/all

I²C Message



Source: K66 Sub-Family Reference Manual, Figure 58-2.

I^2C Principles

- SCL & SDA are pulled high by pull-up resistors.
- ACK bit is set by receiving device (active low for acknowledged).
- Master starts with address transmission – if target exists, it will respond with ACK bit (active low).
- Controller then continues in either transmit or receive mode (according to the read/write bit it sent), and the target continues in the complementary mode (receive or transmit, respectively).
- Bits are sent most significant bit first.

FRDM-K66F I²C

- 4 I²C modules
- Address match wakeup in low-power modes
- SMBus support
- DMA support
- Functions to use:
  - `BOARD_Accel_I2C_Receive`
  - `BOARD_Accel_I2C_Send`

Accelerometer & Magnetometer FXOS8700CQ

- FXOS8700CQ
- Sensor placed directly on FRDM-K66 development board
- Datasheet
  - See 10.1. I^2C interface
  - See 14. Register description
- 3-axis linear accelerometer + 3-axis magnetometer combined into a single package

FXOS8700CQ I²C Connection

- FXOS8700CQ connected to I²C bus and 2 GPIO pins

FXOS8700CQ	K66F Connection
SCL	PTD8/I2C0_SCL
SDA	PTD9/I2C0_SDA
INT1	PTC17
INT2	PTC13

Source: FRDM-K66F Development Platform User's Guide, Table 6.

Seminar task

- Create an application that reads accelerometer output data registers.
- Print register values into console.
- Bonus
  - Calculate tilt angle from received values.

Step-by-step guide

1. Download template from study materials in IS.
2. Look at `initializeAccel` function implementation, pins routing & peripherals.
   - Check if everything is set-up correctly.
3. Read values from sensors in main while loop.
4. Use functions:
   - `BOARD_Accel_I2C_Receive`
   - `BOARD_Accel_I2C_Send`
5. For now, ignore `setupOrientationDetection` functions, it will be used in homework.

Homework – Orientation Detection

- Create an application that detects orientation of the board (the same way as mobile phones do).
- Use the feature of the sensor – do not calculate it in the MCU from XYZ register values.
- Use interrupt from sensor.
- Print current orientation into console when orientation of the board changes.
- Use provided template, write your code into function `setupOrientationDetection`.

MUNI
FACULTY
OF INFORMATICS