ASCII Reader

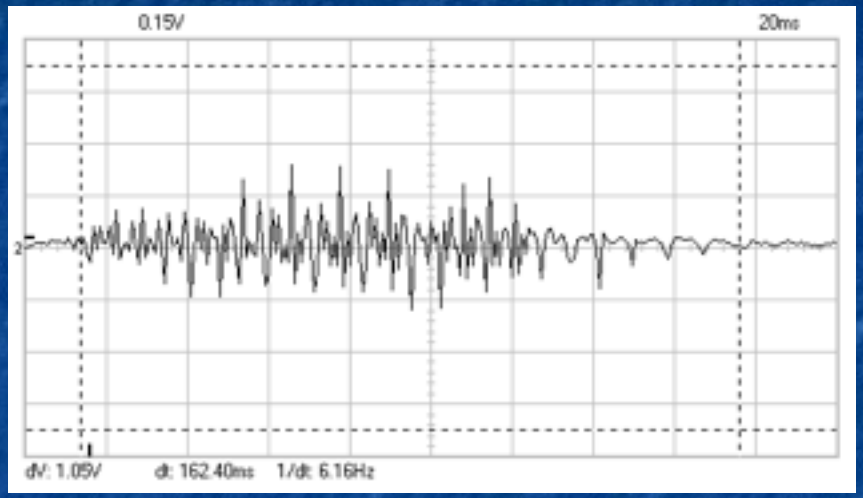
Cristescu Ioana

9th May, 2009

Outline

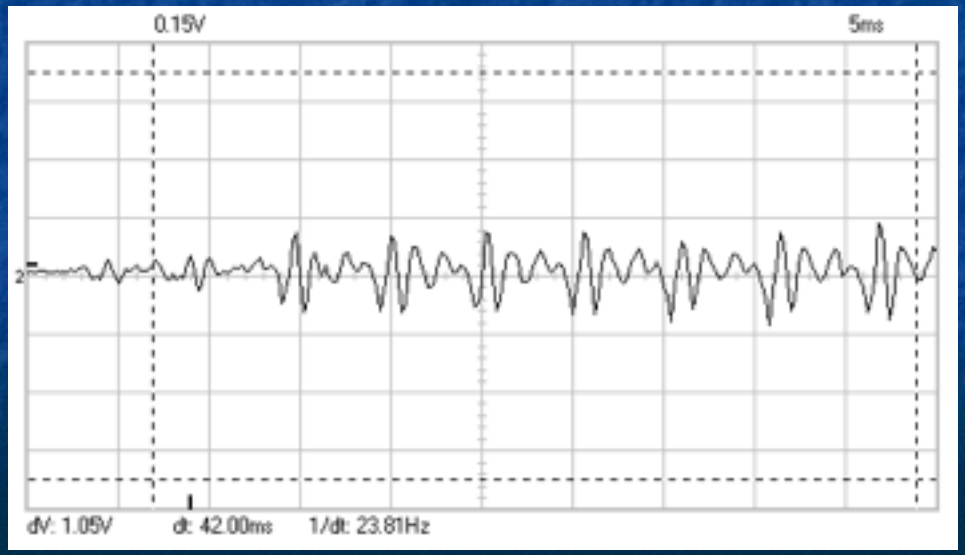
- Letters
- Why microcontroller ?
- Ascii reader:
  - Block diagram
  - Components
  - Implementation details
- Conclusions

Letters

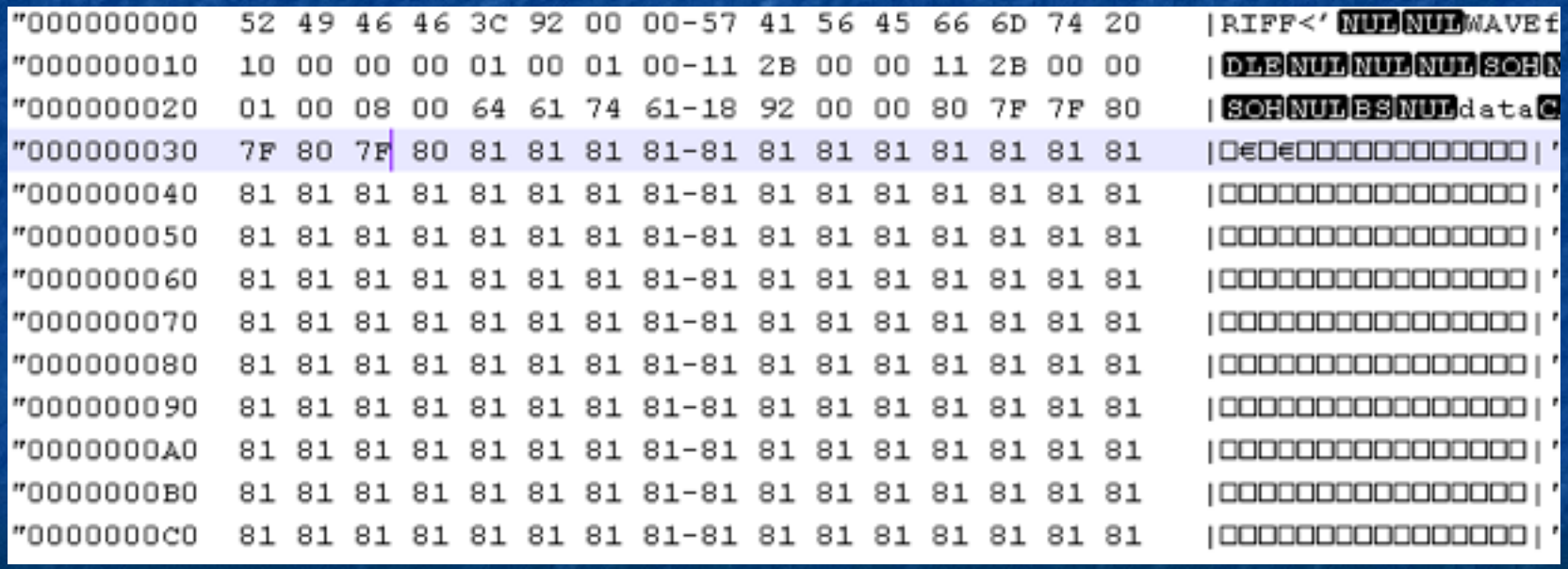


- Sampling frequency: 11.025KHz
- Time between 2 samples: 0.1ms

- Time for letter A pronunciation: 162.40 ms
- Samples needed: 1640



- Letter A



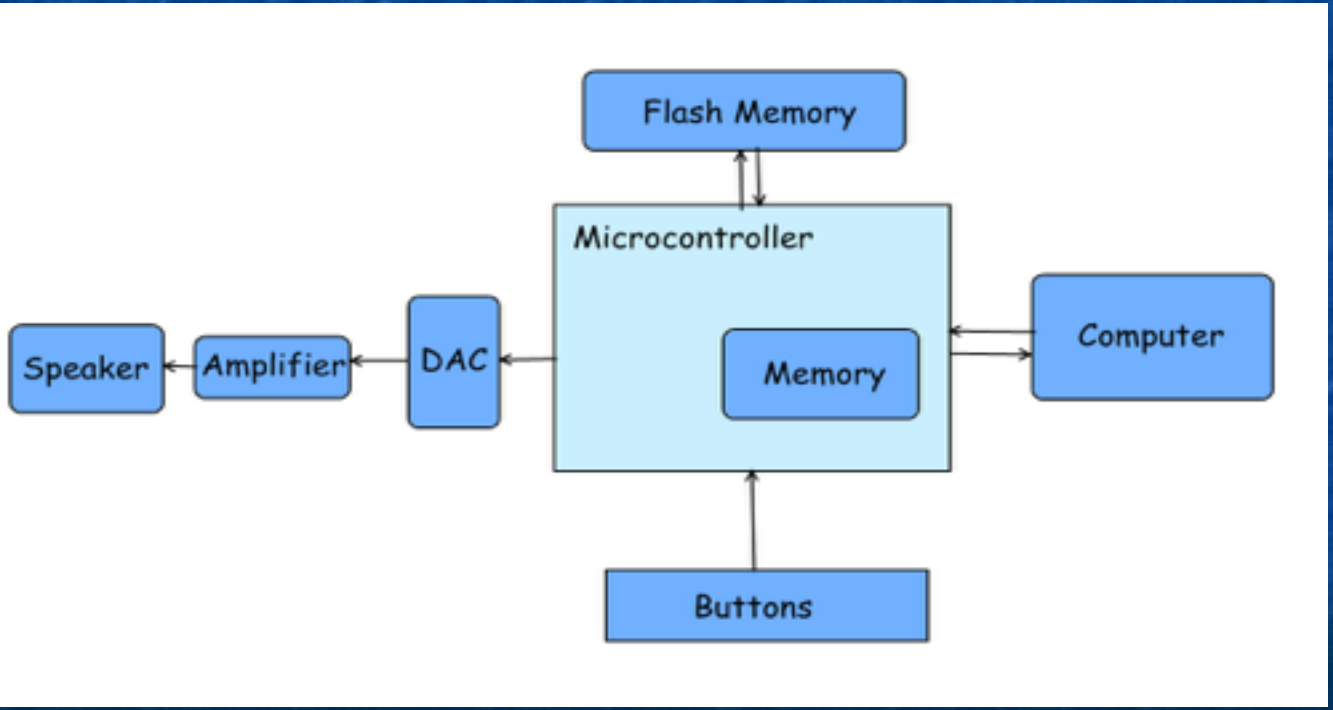
```
"000000000  52 49 46 46 3C 92 00 00-57 41 56 45 66 6D 74 20   |RIFF<' NUL NULWAVEf
"000000010  10 00 00 00 01 00 01 00-11 2B 00 00 11 2B 00 00   | DLE NUL NUL NUL SOH N
"000000020  01 00 08 00 64 61 74 61-18 92 00 00 80 7F 7F 80   | SOH NUL BS NULdataC
"000000030  7F 80 7F 80 81 81 81 81-81 81 81 81 81 81 81 81   |□€□€□□□□□□□□□□□□|'
"000000040  81 81 81 81 81 81 81 81-81 81 81 81 81 81 81 81   |□□□□□□□□□□□□□□□□|'
"000000050  81 81 81 81 81 81 81 81-81 81 81 81 81 81 81 81   |□□□□□□□□□□□□□□□□|'
"000000060  81 81 81 81 81 81 81 81-81 81 81 81 81 81 81 81   |□□□□□□□□□□□□□□□□|'
"000000070  81 81 81 81 81 81 81 81-81 81 81 81 81 81 81 81   |□□□□□□□□□□□□□□□□|'
"000000080  81 81 81 81 81 81 81 81-81 81 81 81 81 81 81 81   |□□□□□□□□□□□□□□□□|'
"000000090  81 81 81 81 81 81 81 81-81 81 81 81 81 81 81 81   |□□□□□□□□□□□□□□□□|'
"0000000A0  81 81 81 81 81 81 81 81-81 81 81 81 81 81 81 81   |□□□□□□□□□□□□□□□□|'
"0000000B0  81 81 81 81 81 81 81 81-81 81 81 81 81 81 81 81   |□□□□□□□□□□□□□□□□|'
"0000000C0  81 81 81 81 81 81 81 81-81 81 81 81 81 81 81 81   |□□□□□□□□□□□□□□□□|'
```

Microcontrollers

- Numerous peripheral devices
- Simplicity
- Fun ☺

ASCII reader

- Block Diagram

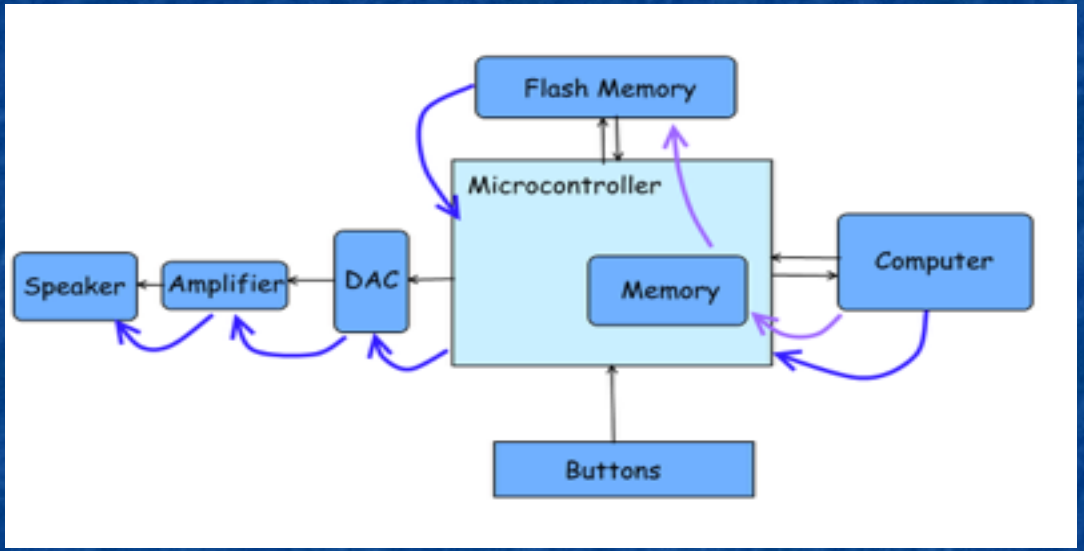


Components

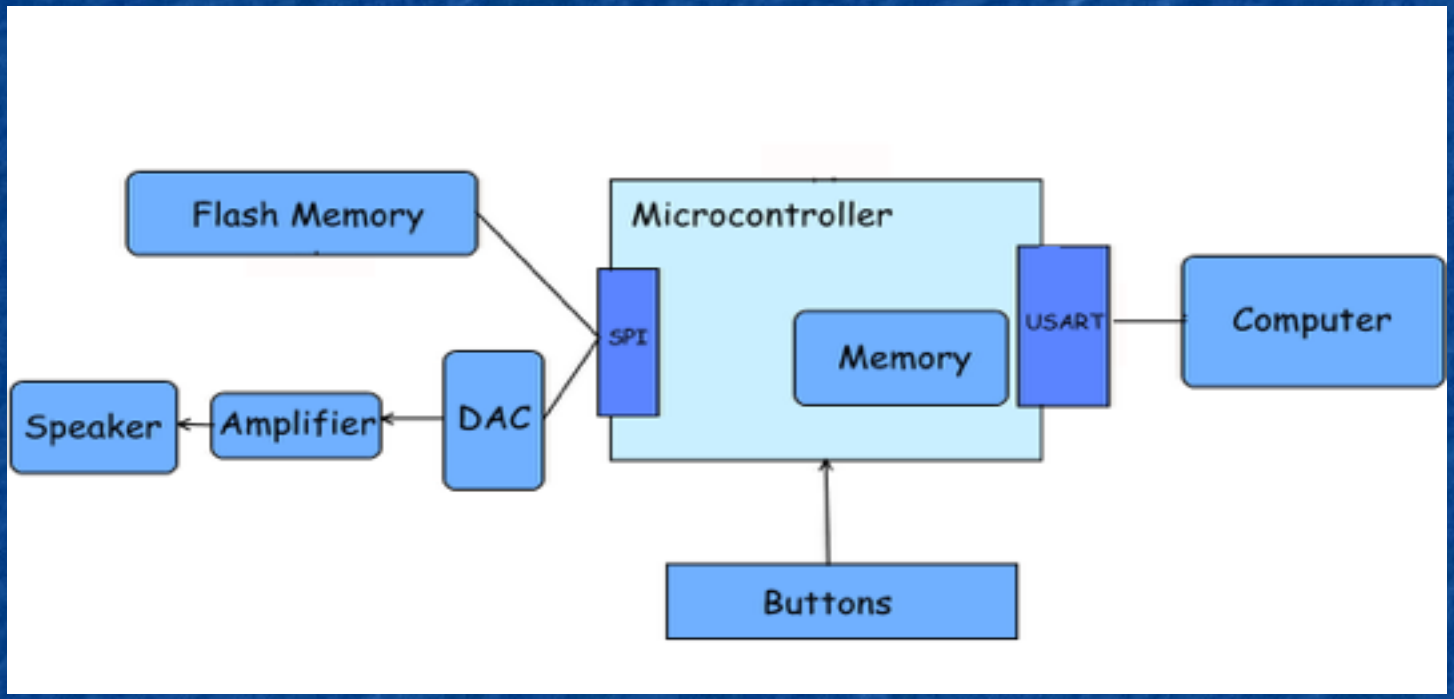
- Divided in 2 asm files.
- Hardware:
  - E.g. microcontroller, flash memory, buttons
- Software:
  - E.g. procedures, interrupt routines

The asm files

- The 2 main objectives:
  - Write files in flash memory
  - Read the data and send it to the speaker



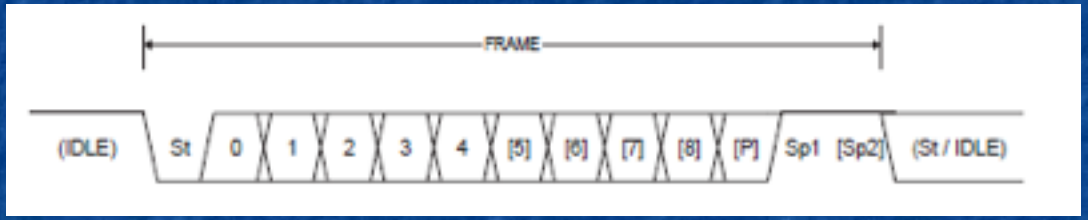
Hardware Components



- Cerebot II board
- Digilent Pmods

USART

- Communication protocol



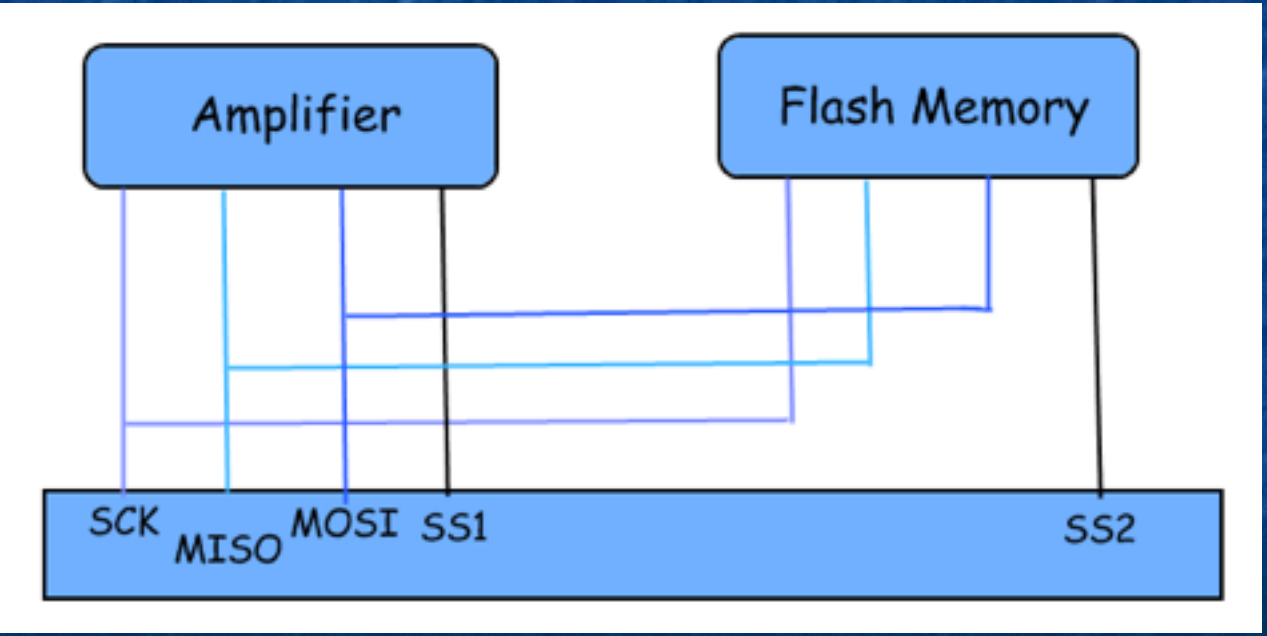
- Sends files (a.txt) and letters (ABAC..) in both direction

SPI

- Computer <-> Flash Memory
  - Instructions, data, control information
- Computer -> Amplifier
  - Data

SPI

- Hardware approach



Flash Memory

- Organization

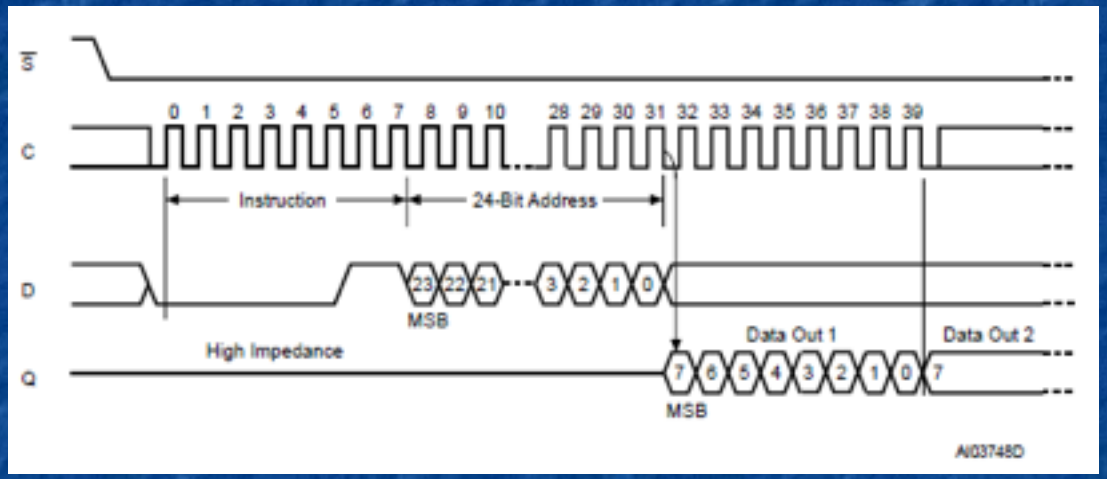
Sector	Address Range	
31	1F0000h	1FFFFFh
30	1E0000h	1EFFFFh
2	020000h	02FFFFh
1	010000h	01FFFFh
0	000000h	00FFFFh

- Instructions
  - Page Programming
  - Read Bytes
  - Erase

Flash Memory

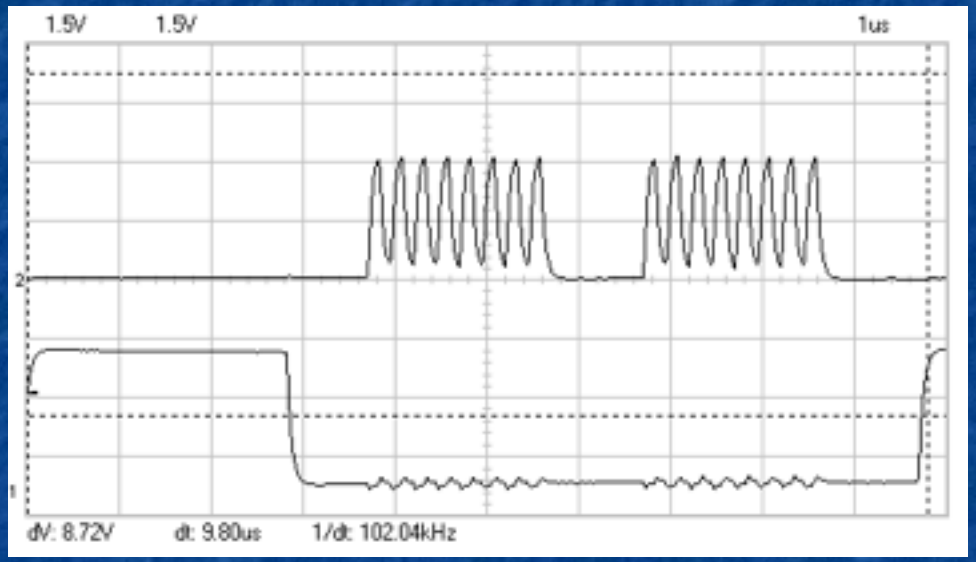
- Instruction Sequence

read data bytes:



- Delays

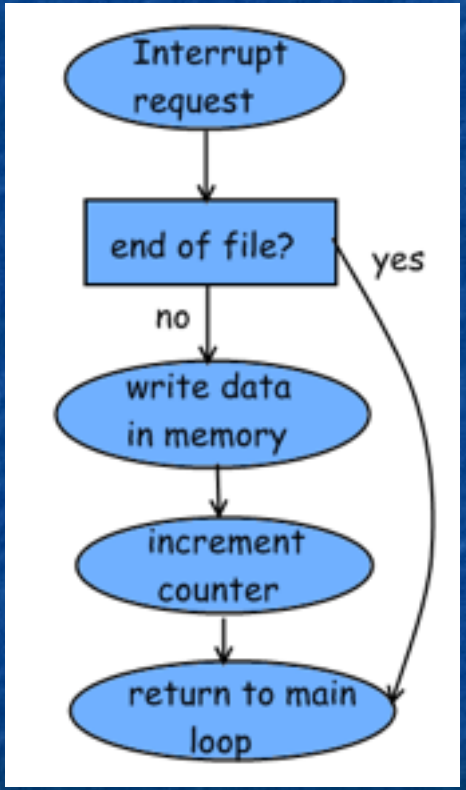
Amplifier



- Chip select
- Data: 12 bits -> 2 bytes

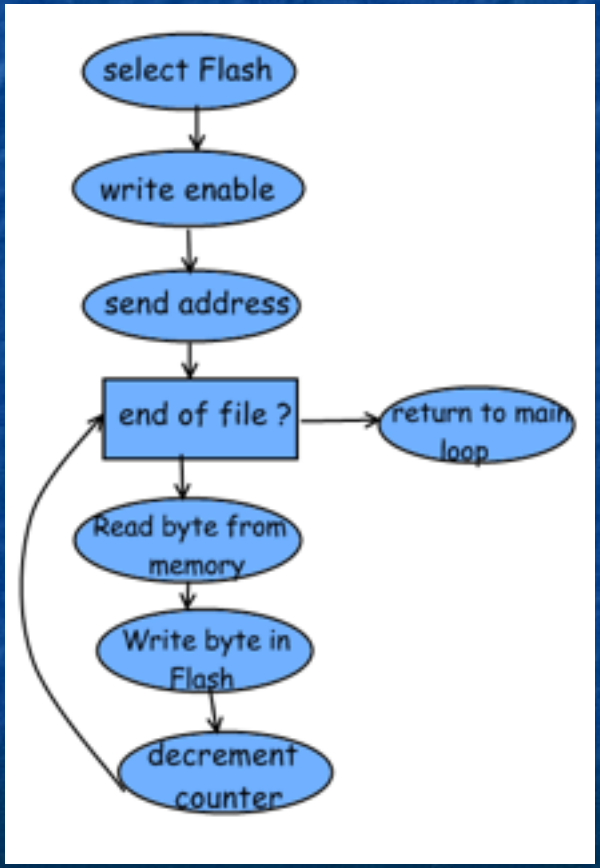
Software implementation

- Receive file from the computer
- Store it in the microcontroller's memory



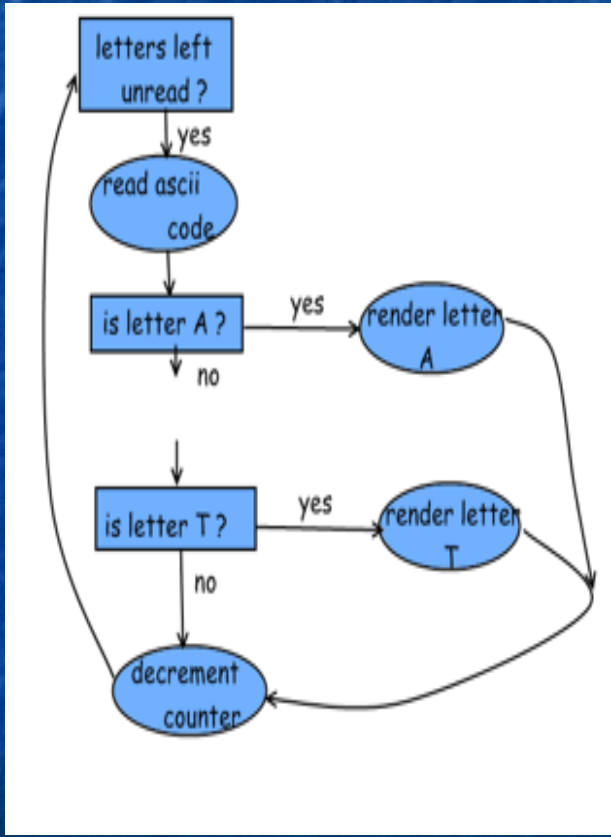
Software Implementation

- Write the content of the microcontroller's memory in the flash memory
- Each letter has it's own sector



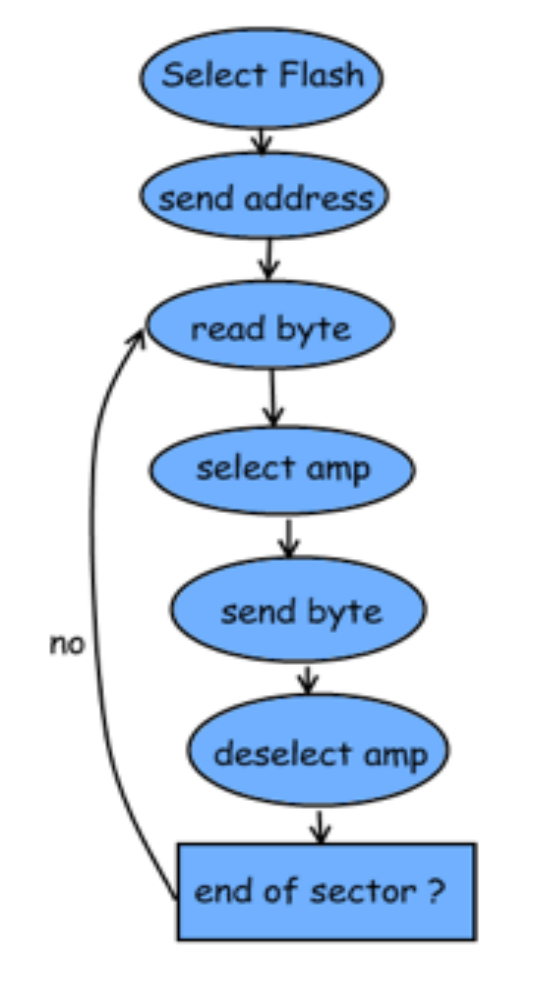
Software Implementation

- The letters sent from the computer are stored in the main memory
- The sector in which the letter is stored in the flash memory is retrieved from the ascii code



Software Implementation

- The communication between flash memory and amplifier



Design choices

- Hardware SPI vs. Software SPI
- Flash -> amplifier vs.

  Flash -> main memory -> amplifier
- Polling vs. interrupts
  - USART
  - SPI

Questions?

Thank You!