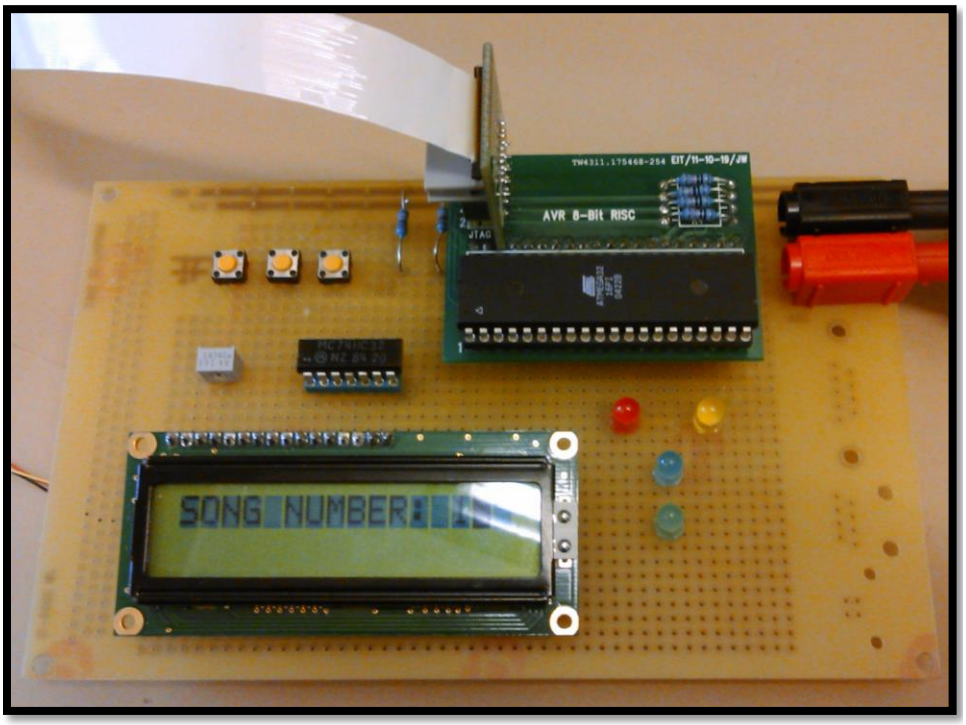


LUNDS UNIVERSITET
Lunds Tekniska Högskola

ATMEGA32
MUSIC PLAYER

Controlled by buttons and accelerometer

The purpose of this project was emulating a music player that can be controlled with external signals. By pressing a button or making a head movement basic operations over the player can be made.

An LCD screen and LEDs indicate the user the reproduction state.

Digital and Analogue Projects

EITF40

09/03/2015

Supervised by: Bertil Lindvall

MARTA LERGO PERDIGUERO

List of contents

1. Abstract

The aim of this project is emulating a music player. The player initializes and starts playing the first song. The user can change the track, stop the reproduction or modify the volume (up or down).

Some LEDs are used to let the user know that the signal for executing a function was received. Each LED corresponds to a different operation:

Red: stop reproduction and go back to the first song.

Blue: modify the volume.

Yellow: go to the next song.

The LCD screen indicates the user the song that is being reproduced or the different operations executed.

All the functions can be executed pressing the different buttons. Only the STOP function is activated when the accelerometer reaches a high value.

2. Hardware specification

The project is based on several hardware components:

Buttons: to receive the signals to run a function that will modify the reproduction. This buttons have a pull-up resistance.

LED lights: help the user to know when the signals are received.

Dot matrix LCD screen, SHARP LM162: it contains two rows with 16 characters each. It receives character codes (8 bits per character) from the microprocessor, transform each character into a 5x7 dot character matrix and displays the character on the LCD screen.

Regulated resistance: used to get the contrast control voltage for the screen.

OR gate: activates the external interruption bit when one of the buttons is pressed.

ATMega32: low power CMOS 8 bit microcontroller. It has four 8 bits I/O ports, which can be used either as standard digital I/O or as other additional functions. The ones used were the AD converter, the external interruptions and the standard digital pins. To program and test the code on the hardware JTAG interface was used. This allows to connect the hardware to a computer via USB.

Accelerometer, MMA7260QT: measures the acceleration of the device itself and returns an output proportional to it. It allows to select among 4 different sensitivities. It also includes a sleep mode.

3. Construction

First of all, the schematics was design. All the peripherals had to be added taking care of the bits used.

The ADC (analog-digital converter) can only be connected to the A port.

The external interruption should be connected to the pins PD2, PD3 or PB2. Instead of using one interruption per button all of them were joined by or gates and connected to the interruption pin.

With this system, every time a button is pressed, the interruption pin is set to 1. When this happens, the microcontroller has to check the source that generated the interruption.

The buttons had to be connected to Vcc, so as to have value 1 in the interruption pin when one of them is pressed. For this reason pull-up resistances had to be added. The microcontroller also has the possibility to activate them internally.

The JTAG is using the pins C3-C6. So they cannot be used for other intentions.

Also, every LED had to be connected to a small resistance and to ground. With this system, they will only be activated when the microcontroller sends a 1.

The accelerometer was attached to a hairband, so the movement to activate the STOP function should be done with the head.

4. Programming

Some details were taking into consideration, to make the program more efficient:

The ADC should not be running permanently, as the other interruptions also should be controlled. Besides, the ADC is very fast and that much speed is not needed. That is why a timer was set. The microcontroller has 3 different ones. The one chosen was timer0, as it is the only one with only 8 bits (no more are needed). The timer was set to generate an interruption every 0.1 seconds more or less. When that happens, the ADC is checked.

The output voltage from the accelerometer is not a really high value, so a small reference voltage for the ADC had to be chosen. ATMega32 has an internal reference voltage of 2.56 V, so this was more convenient than taking Vcc.

When the program initializes some messages are displayed. Also after a function is executed. During this time, the interruptions are disabled. This is to avoid incomplete messages on the screen.

Find the table with the description of every designed function.

`void InitINT()`	Initializes the interruptions.
`void ISR(INT1_vect)`	External interruption function. In this case it is run when any of the buttons is pressed.
`void InitADC()`	Initializes the analog-digital converter (ADC).
`unsigned int ReadADC()`	Starts a conversion.
`void init_LCD()`	Initializes the LCD screen.
`void LCD_cmd(unsigned char cmd)`	Sends a command to the display.
`void LCD_write(unsigned char data)`	Writes a character on the display.
`void clear_display()`	Clears display and returns cursor to the home position.
`void Entry_mode_set(unsigned char I_D, unsigned char S)`	Sets cursor move direction and specifies shift of the display.
`void display_control(unsigned char D, unsigned char C, unsigned char B)`	Sets On/Off display, cursor On/Off and blink of cursor position character.
`void cursor_display_shift(unsigned char S_C, unsigned char R_L)`	Sets cursor-move or display-shift and shift direction.
`void function_set(unsigned char DL,unsigned char N,unsigned char F)`	Sets interface data length, number of display lines and character font.
`void print_string(char word[])`	Prints a string on the screen.
`void print_integer(int number)`	Prints an integer on the screen.
`void initTimer0()`	Initializes the timer0.
`void ISR(TIMER0_COMP_vect)`	Timer0 interruption function.
`ISR(ADC_vect)`	ADC interruption function. It runs when the conversion is finished.

5. Conclusions and comments

The project concluded in a music player with several reproduction possibilities. The sensitivity for the changes in reproduction is good. The system detects instantly when buttons are pressed and the movement for the STOP function is easily recognized.

Some possibilities for developing more or improving this project would be:

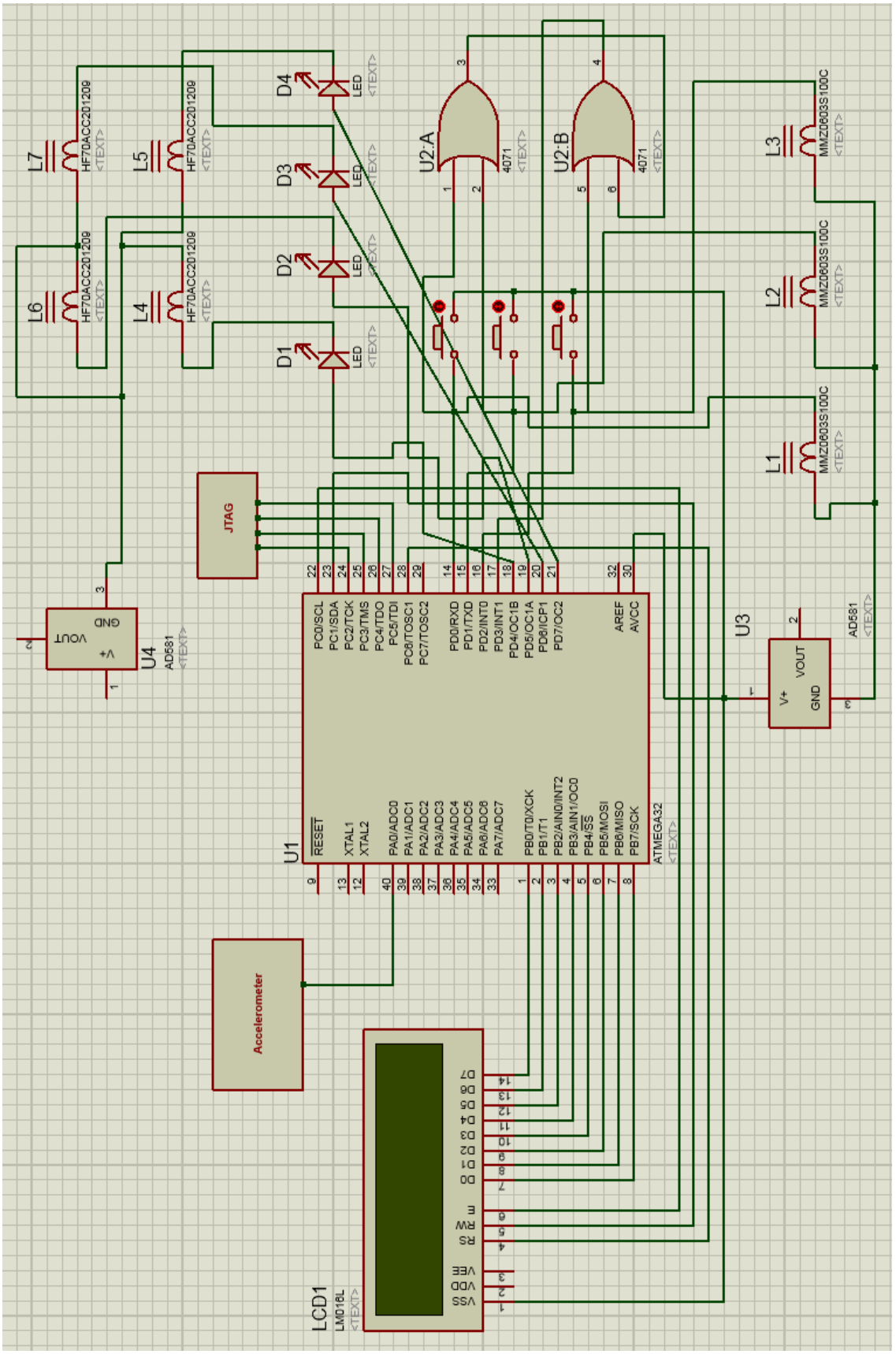
-Adding real music tracks.

-Complement it with more movement possibilities.

-Adding custom characters to the display, for instance music characters.

6. Attachments

-Schematics



-Datasheets

-ATMega32: www.atmel.com/Images/doc2503.pdf

-Display: www.home.zcu.cz/~dudacek/manuals/lm.pdf

-Accelerometer:

http://www.datasheetcatalog.com/datasheets_pdf/M/M/A/7/MMA7260Q.shtml