Ministerul Educaţiei şi Tineretului al Republicii Moldova

Universitatea Tehnică a Moldovei

REPORT

laboratory work #2

Embedded Systems

A efectuat:

St. gr. FAF 141 Schidu Luca

A verificat:

Lect . supp Bragrarenco Andrei

Chisinau 2016

Topic: Make a user interface(button) with a LED.

Objectives: Get familiar with the push button and LED components and learning how to connect them with ATMega32.

Tasks: Write a program for the microcontroller, that will turn on and off a LED, connected to it, by a push-button;

Short Theory:

Atmega32 MCU - The high-performance, low-power Atmel 8-bit AVR RISC-based microcontroller combines 32KB of programmable flash memory, 2KB SRAM, 1KB EEPROM, an 8-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device supports throughput of 16 MIPS at 16 MHz and operates between 4.5-5.5 volts. By executing instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

A light-emitting diode (LED) is a two-lead semiconductor light source. It is a p–n junction diode, which emits light when activated. When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor.

A **push-button** (also spelled **pushbutton**) or simply **button** is a simple switch mechanism for controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal.The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, though even many un-biased buttons (due to their physical nature) require a spring to return to their un-pushed state. Different people use different terms for the "pushing" of the button, such as **press**, **depress**, **mash**, **hit**, and **punch**.

All this components are connected and all commands are given through virtual terminal. On command 0 LED is off, on command 1 LED is on.

Working process:

First we wrote main.c file, compiled it, and checked if it works. Next we created the files for the LED and the PUSH-BUTTON, for the LED file I've created Three function.

Init(); - This function sets the DDR of where the LED is connected to 1 (output);

ledOn(); - This function set's the first pin of the port A to 1, meaning the current can pass to the LED, so it will turn on.

LefOff(); - This function sets the first pin of the port A to 0, meaning the current cannot pass to the LED, so it will turn off.

Functions for the push-button();

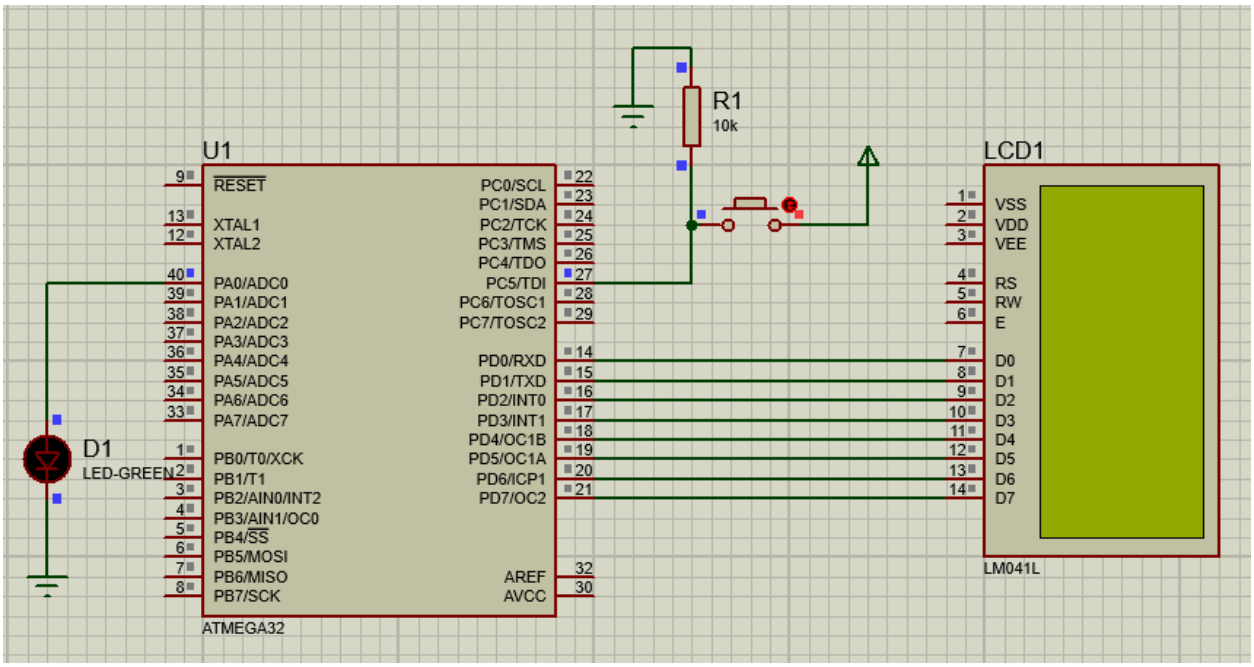
InitButton(); = this function sets the DDR of where the button is connected to 0 (input).

IsPressed() - this function check if the first pin of the PORTC is 0 or 1.

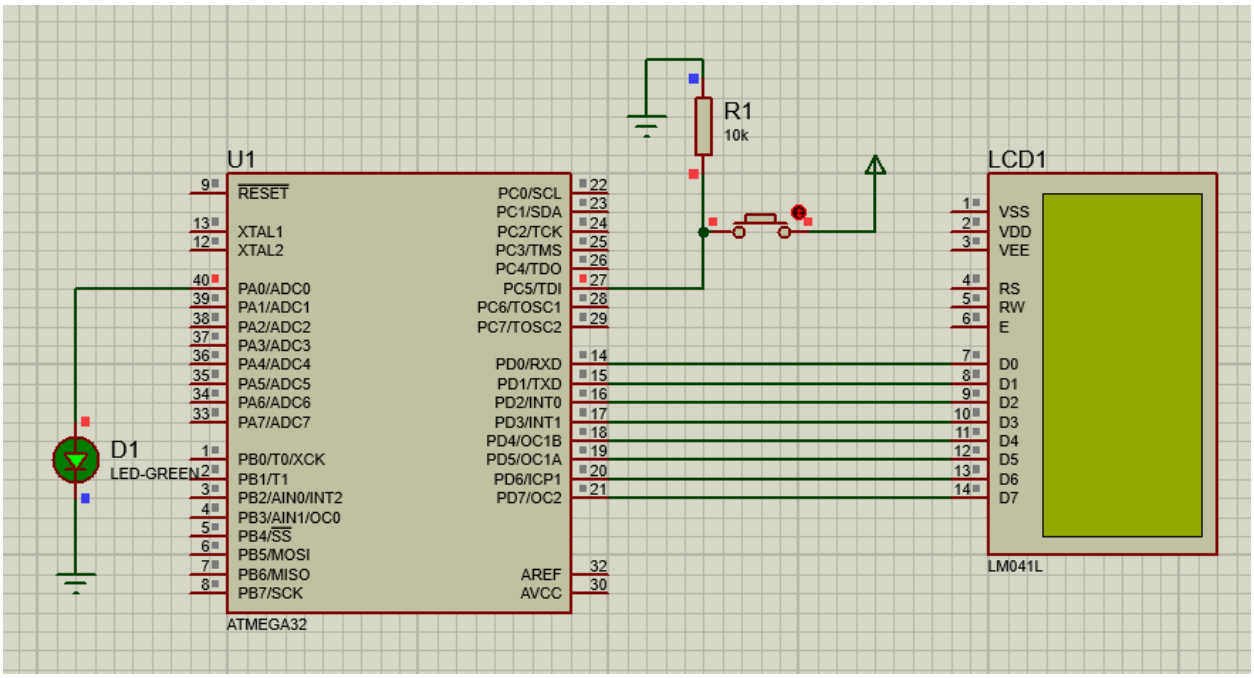
The main function runs in an infinite loop, with a delay of 100 ms. In the loop I check if the button is pressed or not, with the function I described above. Depending on the situation I either turn on or off the LED.

In proteus I made the coresponding schematic, and linked it to the .HEX file that was generated in the process of building the project. I emulated it and here's the result.

LED OFF


R1
10k
U1
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
ATMEGA32
D1
LED-GREEN
LCD1
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
LM041L

LED ON


R1
10k
U1
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
ATMEGA32
D1
LED-GREEN
LCD1
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
LM041L

Conclusion:

In this laboratory work I learned about different components that we can use to integrate into ATMega32 microcontroller. Push-button is kind of a switch that can be in two states – pushed(1) and released(0). LED is a two-lead semiconductor light source. I've learned how to connect them to the microcontroller and also make them interact. (By pushing the button to light on the LED).

Appendix

main.c

```
/*
 * main.c
 *
 * Created: 10/4/2016 11:08:42 AM
 *  Author: schid
 */

#include "led.h"
#include "uart_studio.h"
#include "button.h"
#include <avr/delay.h>


int main() {

	init();
	initLed();

	while(1) {
		_delay_ms(100);
		if(isPressed()) {
			ledOn();
		} else {
			ledOff();
		}
	}



	return 0;
}
```

led.h

```
/*
 * led.h
 *
 * Created: 10/2/2016 1:51:57 PM
 *  Author: schid
 */


#ifndef LED_H_
#define LED_H_
```

```
#include <avr/io.h>

void initLed();
void ledOn();
void ledOff();



#endif /* LED_H_ */
```

led.c

```
/*
 * led.c
 *
 * Created: 10/2/2016 1:52:30 PM
 *  Author: schid
 */
#include "led.h"

void initLed() {
	DDRA |= (1 << PORTA0);
}

void ledOn() {
	PORTA |= (1 << PORTA0);
}

void ledOff() {
	PORTA &= ~(1 << PORTA0);
}
```

button.h

```
/*
 * button.h
 *
 * Created: 10/2/2016 1:43:21 PM
 *  Author: schid
 */


#ifndef BUTTON_H_
#define BUTTON_H_
#include <avr/io.h>

int isPressed();
void init();
```

```
#endif /* BUTTON_H_ */
```

button.c

```
/*
 * button.c
 *
 * Created: 10/2/2016 1:43:03 PM
 *  Author: schid
 */
#include "button.h"

void init() {
	DDRC &= ~(1 << PORTC5) ;
}

int isPressed() {
	return PINC & (1<<PORTC5);
}
```