Ministry of Education of Moldova
Technical University of Moldova
Faculty "Computers, Informatics and Microelectronics"

REPORT

Laboratory work No.2
On Embedded systems

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

Performed by st. gr. FAF-141 : Botnari Nicolae

Verified by : Andrei Bragarenco

Chişinău 2016

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

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

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

Theory:

Microcontroller

A microcontroller (or MCU for microcontroller unit) is a small computer on a single integrated circuit. In modern terminology, it is a System on a chip or SoC. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Program memory in the form of Ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

Some microcontrollers may use four-bit words and operate at frequencies as low as 4 kHz, for low power consumption (single-digit milliwatts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

Led:

A light-emitting diode (LED) is a two-lead semiconductor light source. It is a p–n junction diode, which emits light when activated.[4] 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.

An LED is often small in area (less than 1 mm2) and integrated optical components may be used to shape its radiation pattern.

Solution:

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.

ledOff();

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.

InitButton();

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

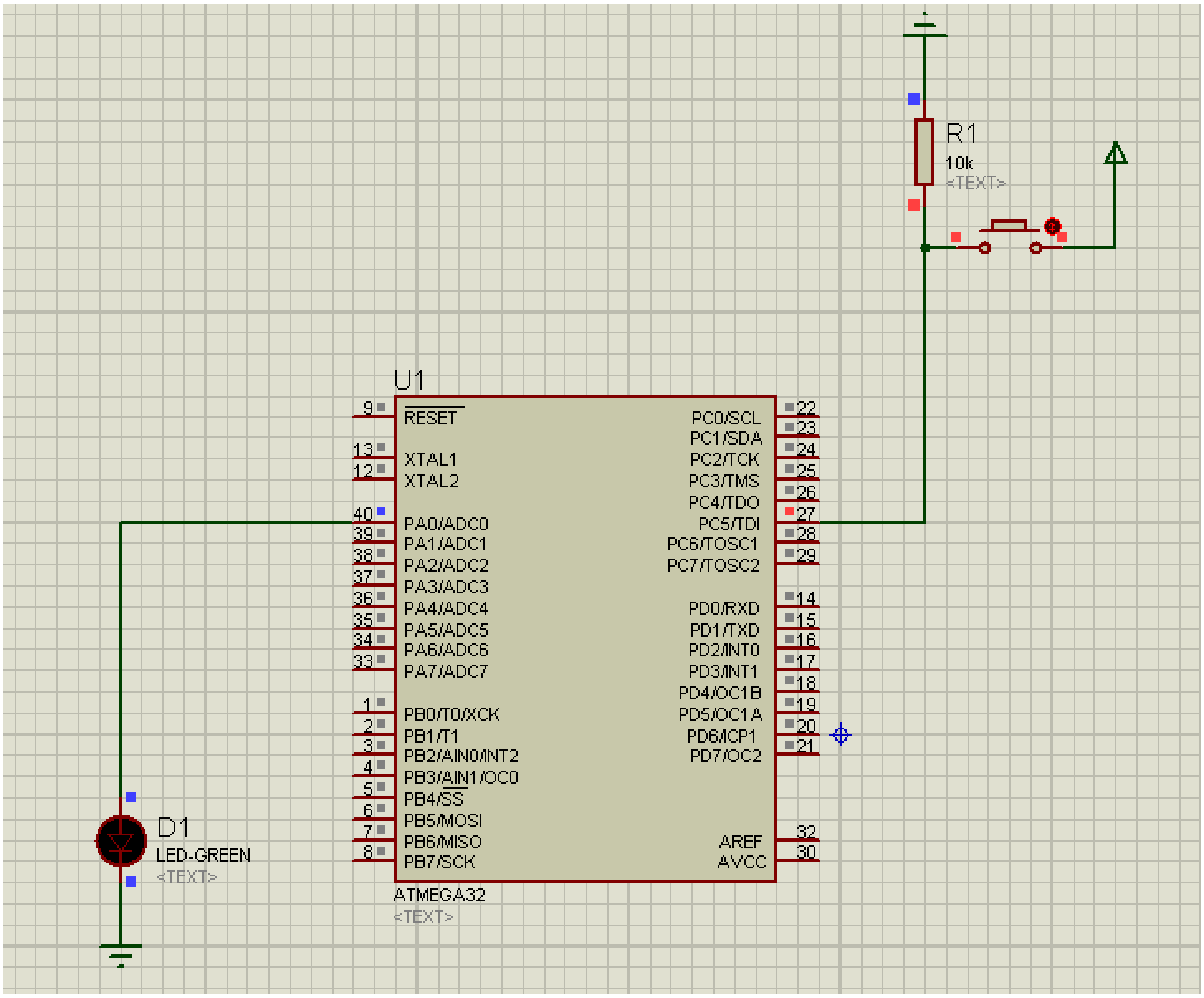
IsPressed()

this function check if the first pin of the PORTC is 0 or 1.

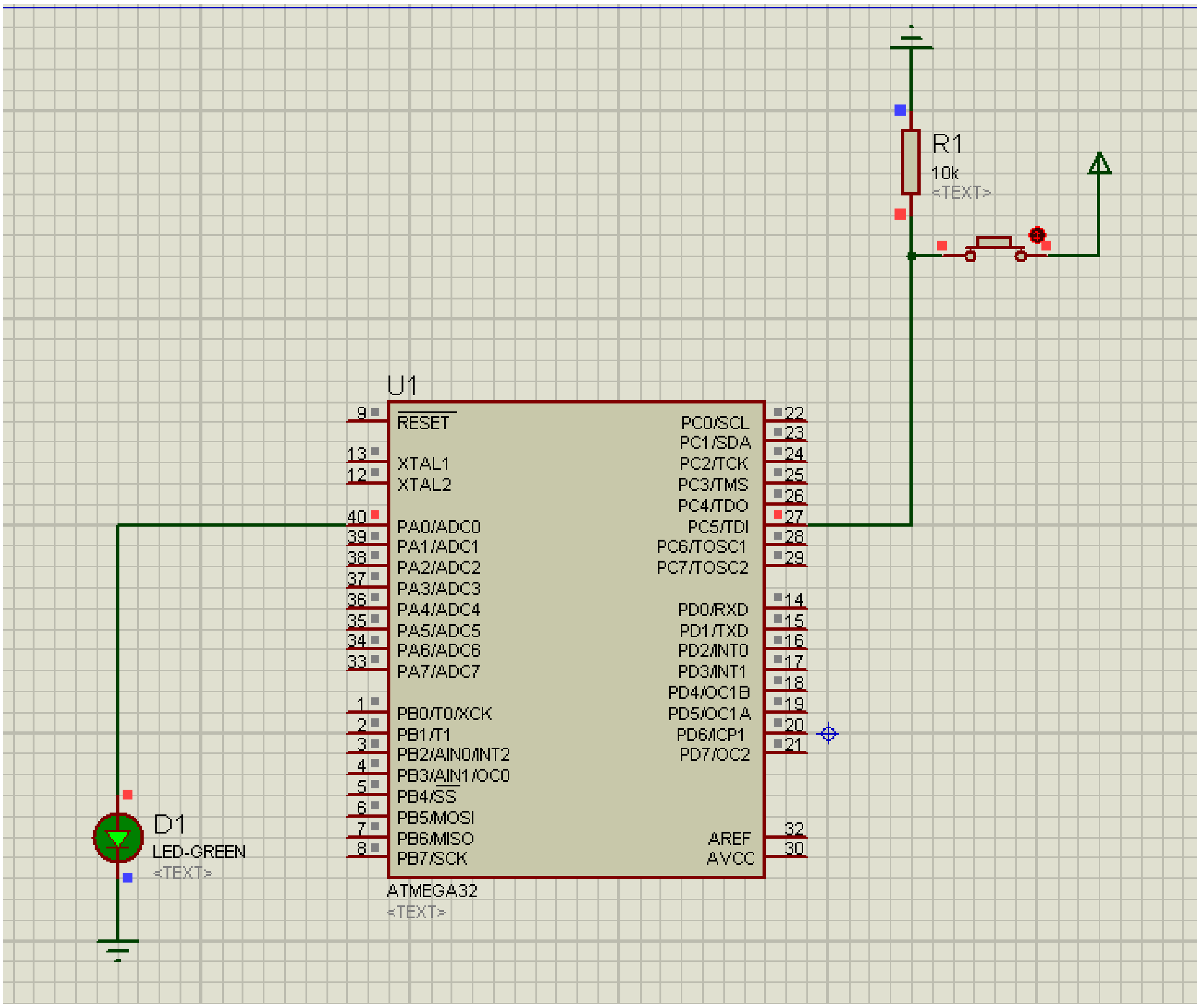
The main function runs in an infinite loop, with a delay of 100 ms. In the loop I check if the button is pressed or not, with the function I described above. Depending on the situation I either turn on or off the LED.

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
<TEXT>
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
9
13
12
40
39
38
37
36
35
34
33
1
2
3
4
5
6
7
8
22
23
24
25
26
27
28
29
14
15
16
17
18
19
20
21
32
30
ATMEGA32
<TEXT>
D1
LED-GREEN
<TEXT>

LED ON


R1
10k
<TEXT>
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
9
13
12
40
39
38
37
36
35
34
33
1
2
3
4
5
6
7
8
22
23
24
25
26
27
28
29
14
15
16
17
18
19
20
21
32
30
ATMEGA32
<TEXT>
D1
LED-GREEN
<TEXT>

Conclusion:

In this laboratory work I learned about different components that we can use to integrate into ATMega32 microcontroller. We used push button as a switch for the LED and I’ve learned how to connect them to the microcontroller and also make them interact.

Appendix:

Main.c

```
#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;
}
```

Button.h

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

int isPressed();
void init();


#endif /* BUTTON_H_ */
```

Button.c

```
#include "button.h"

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

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

Led.h

```c
#ifndef LED_H_
#define LED_H_
#include <avr/io.h>

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

#endif /* LED_H_ */
```

Led.c

```c
#include "led.h"

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

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

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