Embended systems
Microcontrollers
Fall 2016
Laboratory Work 3

Vasile Schidu

January 21, 2017

Date Performed:	October 02, 2016
Partners:	Vasile Schidu
Instructor:	Bragrarenco Andrei

Introduction

Topic: Converting Analog to Digital signal. Connecting temperature sensor to MCU and display temperature to display

Objectives:

1. Initiation in Analog to digital converter (ADC)
2. Connecting Temperature sensor to MCU

Tasks: Write driver for ADC and LM20 Temperature sensor. ADC will transform Analog to Digital data. LM20 driver will use data from ADC to transform to temperature regarding to this sensor parameters. Also use push button to switch between metrics Celsius, Fahrenheit and Kelvin.

ADC

Definition Analog-to-digital conversion is an electronic process in which a continuously variable (analog) signal is changed, without altering its essential content, into a multi-level (digital) signal. The input to an analog-to-digital converter (ADC) consists of a voltage that varies among a theoretically infinite number of values. Examples are sine waves, the wave forms representing human speech, and the signals from a conventional television camera. The output of the ADC, in contrast, has defined levels or states. The number of states is almost always a power of two – that is, 2, 4, 8, 16, etc. The simplest digital signals have only two states, and are called binary. All whole numbers can be represented in binary form as strings of ones and zeros.

How to use Micro-controllers 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 micro-controllers are common, integrating analog components needed to control non-digital electronic systems.

How it works Even though there is a large number of different types of micro-controllers and even more programs created for their use only, all of them have many things in common. Thus, if you learn to handle one of them you will be able to handle them all. A typical scenario on the basis of which it all functions is as follows:

1. Power supply is turned off and everything is still the program is loaded into the micro-controller, nothing indicates what is about to come
2. Power supply is turned on and everything starts to happen at high speed! The control logic unit keeps everything under control. It disables all other circuits except quartz crystal to operate. While the preparations are in progress, the first milliseconds go by
3. Power supply voltage reaches its maximum and oscillator frequency becomes stable. SFRs are being filled with bits reflecting the state of all circuits within the micro-controller. All pins are configured as inputs. The overall electronics starts operation in rhythm with pulse sequence. From now on the time is measured in micro and nanoseconds.
4. Program Counter is set to zero. Instruction from that address is sent to instruction decoder which recognizes it, after which it is executed with immediate effect.
5. The value of the Program Counter is incremented by 1 and the whole process is repeated several million times per second.

Special Function Registers

Special function registers are part of RAM memory. Their purpose is prede- fined by the manufacturer and cannot be changed therefore. Since their bits are physically connected to particular circuits within the micro-controller, such as A/D converter, serial communication module etc., any change of their state directly affects the operation of the micro-controller or some of the circuits. For example, writing zero or one to the SFR controlling an input/output port causes the appropriate port pin to be configured as input or output. In other words, each bit of this register controls the function of one single pin.

Program Counter

Program Counter is an engine running the program and points to the memory address containing the next instruction to execute. After each instruction execution, the value of the counter is incremented by 1. For this reason, the program executes only one instruction at a time just as it is written. However the value of the program counter can be changed at any moment, which causes a "jump" to a new memory location. This is how subroutines and branch instructions are executed. After jumping, the counter resumes even and monotonous automatic counting +1, +1, +1. . .

Resourses

Short Theory: Proteus developed by Labcenter Electronics, is a software with which you can easily generate schematic captures, develop PCB and simulate microprocessor. It has such a simple yet effective interface that it simplifies the task required to be performed. This one aspect has attracted many users to select this tool amongst many others offering the same services. Atmel® Studio 6 is the integrated development platform (IDP) for developing and debugging Atmel ARM® Cortex®-M and Atmel AVR® microcontroller (MCU) based applications. The Atmel Studio 6 IDP gives you a seamless and easy-to-use environment to write, build and debug your applications written in C/C++ or assembly code. A microcontroller (sometimes abbreviated μC, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications. In computing, a device driver (commonly referred to as a driver) is a computer program that operates or controls a particular type of device that is attached to a computer. A driver provides a software interface to hardware devices, enabling operating systems and other computer programs to access hardware functions without needing to know precise details of the hardware being used.

About stdio library: Input and Output operations can be performed using theCStandardInput andOutput Library (cstdio, known asstdio.hin the C language). This library uses what are calledstreamsto operate with physical devices such as keyboards, printers, terminals or with any other type of files supported by the system. Streams are an abstraction to interact with these in an uniform way. All streams have similar properties independently of the individual characteristics of the
physical media they are associated with. Streams are handled in thecstdiolibrary as pointers toFILEobjects. A pointer to aFILEobject uniquely identifies a stream, and is used as a parameter in the operations involving that stream.

Atmel® microcontrollers: Atmel® microcontrollers (MCUs) deliver a rich blend of efficient integrated designs, proven technology, and groundbreaking innovation that is ideal for today's smart, connected products. In this era of the Internet of Things (IoT), microcontrollers comprise a key technology that fuels machine-to-machine (M2M) communications. Building on decades of experience and industry leadership, Atmel offers proven architectures that are optimized for low power, high-speed connectivity, optimal data bandwidth, and rich interface support. By using our wide variety of configuration options, developers can devise complete system solutions for all kinds of applications. Atmel microcontrollers can also support seamless integration of capacitive touch technology to implement buttons, sliders, and wheels (BSW). In addition, Atmel MCUs deliver wireless and security support. No matter what your market or device, Atmel offers a compelling solution that is tailored to your needs—today and tomorrow. Atmel is a global industry leader in the design and manufacture of microcontrollers and related system solutions, including capacitive touch solutions, advanced logic, mixed-signal, nonvolatile memory, and radio frequency (RF) components. Leveraging one of the industry's broadest intellectual property technology portfolios and backed by a comprehensive ecosystem, Atmel MCU products enable designers to develop complete solutions for industrial, consumer, security, communications, computing, and automotive markets. Developers have the option of combining Atmel microcontrollers with industry-leading Atmel touch technology. Atmel technology for touchscreens and fixed-function buttons, sliders and wheels provides a rich user experience with unparalleled performance, while minimizing power consumption.

ATmega328

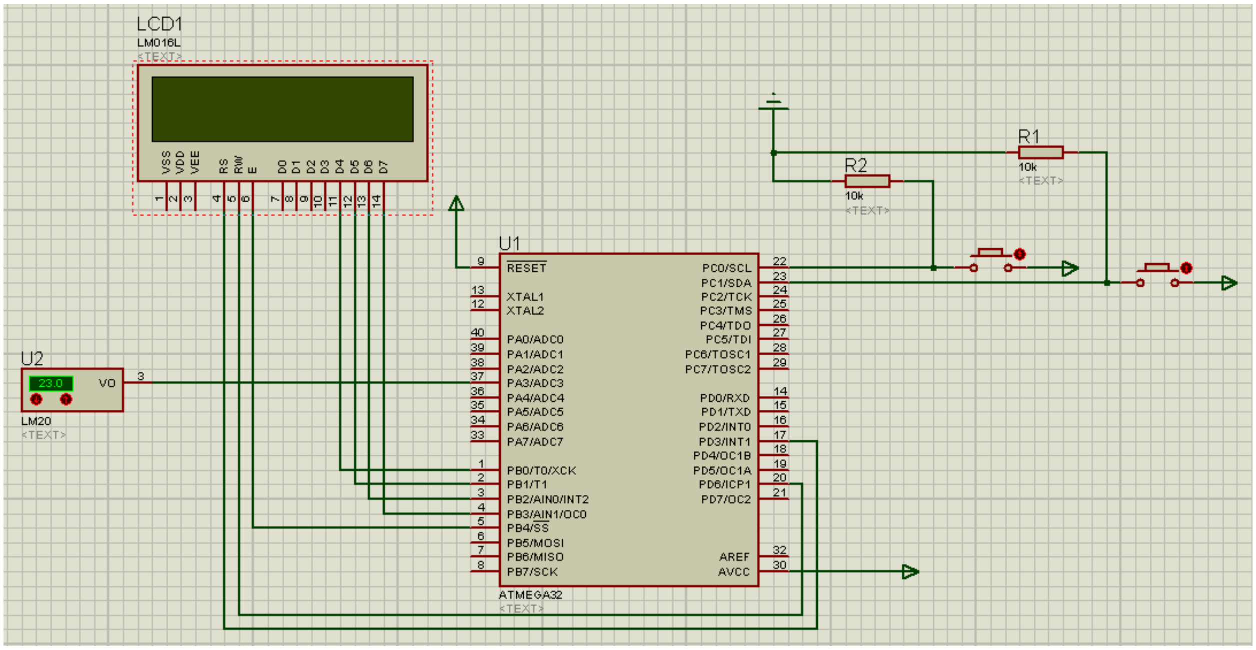
The high-performance Atmel 8-bit AVR RISC-based microcontroller combines 32KB ISP flash memory with read-while-write capabilities, 1KB EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts,serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

Atmel Studio: Atmel Studio 7 is the integrated development platform (IDP) for developing and debugging Atmel® SMART ARM®-based and Atmel AVR® microcontroller (MCU) applications. Studio 7 supports all AVR and Atmel SMART MCUs. The Atmel Studio 7 IDP gives you a seamless and easy-to-use environment to write, build and debug your applications written in C/C++ or assembly code. It also connects seamlessly to Atmel debuggers and development kits. Additionally, Atmel Studio includes Atmel Gallery, an online apps store that allows you to extend your development environment with plug-ins developed by Atmel as well as by third-party tool and embedded software vendors. Atmel Studio 7 can also able seamlessly import your Arduino sketches as C++ projects, providing a simple transition path from Makerspace to Marketplace.

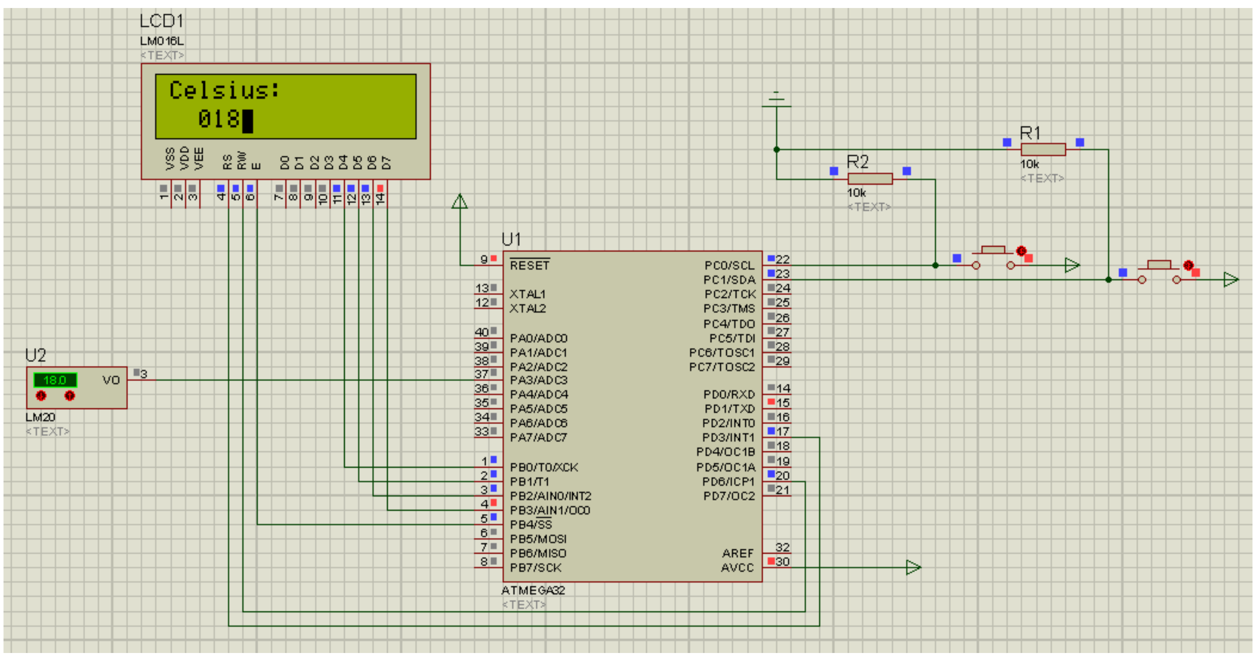
Solution

Proteus Scheme Let's take a look on this scheme. So we have Microcontroler **ATMEGA32** This microcontroler is composed from 4 ports each of thie have 8 pins.

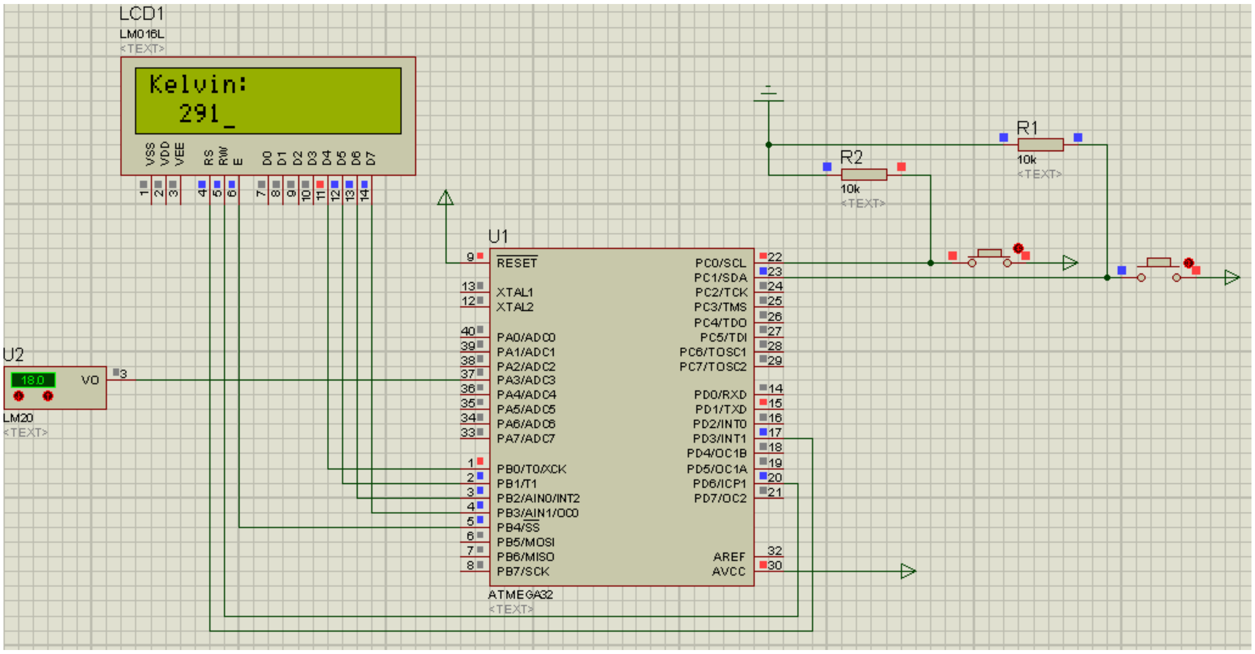
I've connected MCU to Virtual Terminal. Because I use only data transmission on one direction (OUTPUT) we need to make sure that our MC Tx is connected to Peripheral Rx. Also I connected LM20 to ADC pin 3 and also connected LCD.



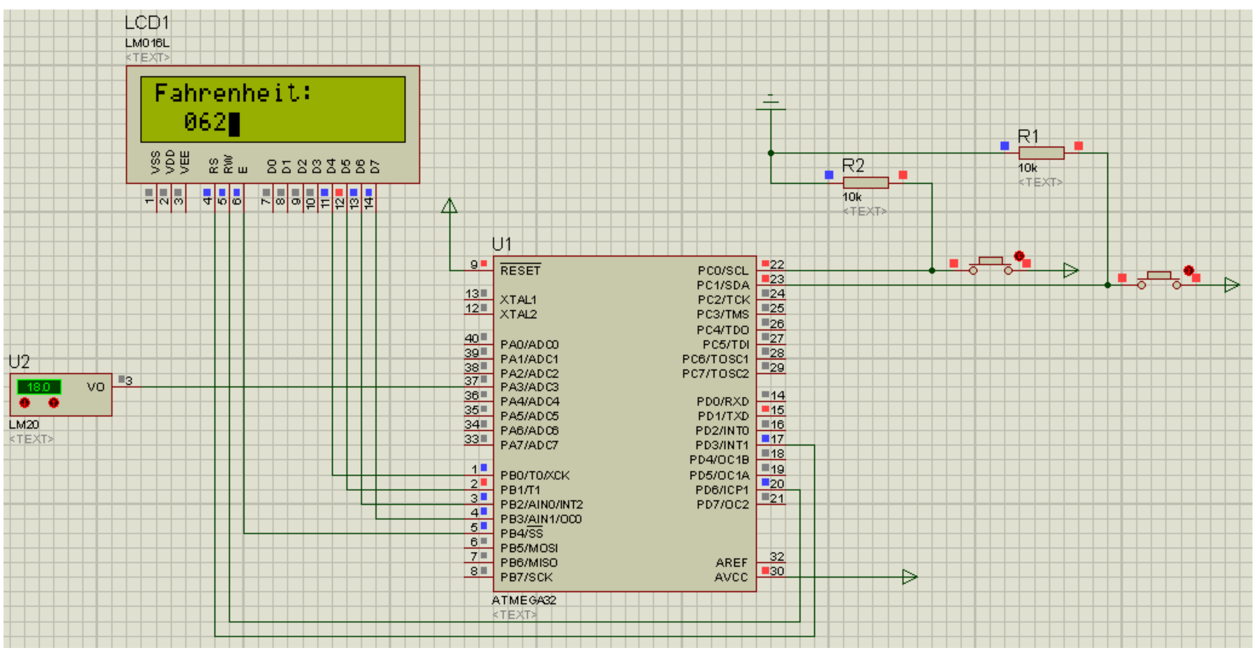
So next lets run our project. As was said if no button is presed the LCD will show temperature in Celsius



When we puch 1st button the LCD shows the temperature in Kelvin



And when we push both buttons the temperature is shown in Fahrenheitt



Conclusion

In this laboratory work I've learned basic concepts of MCU programming in C language and building a simple printed circuit using Proteus. I've implemented a driver for LM20 and using this driver I've did a small embedded system that get the temperature from LM20 and displays it in a virtual terminal and also on LCD.