TOSHKENT SHAHRIDAGI INHA UNIVERSITETI
INHA UNIVERSITY IN TASHKENT

Embedded System

Requirements Specification and Analysis

Team name:

CD Projekt Green

Team Members:

Oybek Khakimjanov	U1610179
Dilmurod Nasibullaev	U1610058
Anvar Abdulsatarov	U1610025

Introduction

There is usually no way around the fact that the time is the most valuable resource in our life. A precise measurement of the time had always been one of the main concerns that software engineers needed to resolve. Numerous methodologies and approaches have been proposed, but the simplest one used today is abstraction of CPU’s clock.

Digital clocks revolutionize the way we live our daily life as it helps people to stick with their schedule. All the microcontrollers have a crystal inside of them which is the heart of clock. This crystal present in all modern computing systems including ATMEGA128 microcontroller that will be used in this project. This crystal generates clock pulses, which is needed for timing calculations. ATMEGA128 usually operates at 14.7456 MHz from that frequency we are going to approximate our timing scale.

This document describes all the requirements and criteria that should be accomplished in our final assignment. A brief overview of functionalities will be discussed in detail.

Components Required:

- Atmel Studio
- SimulIDE
- Circuit scheme

E

General Requirements

In this project, the hearth of this **digital alarm clock** is AtMega128 microcontroller.

The digital clock should be able to operate in the following three modes:

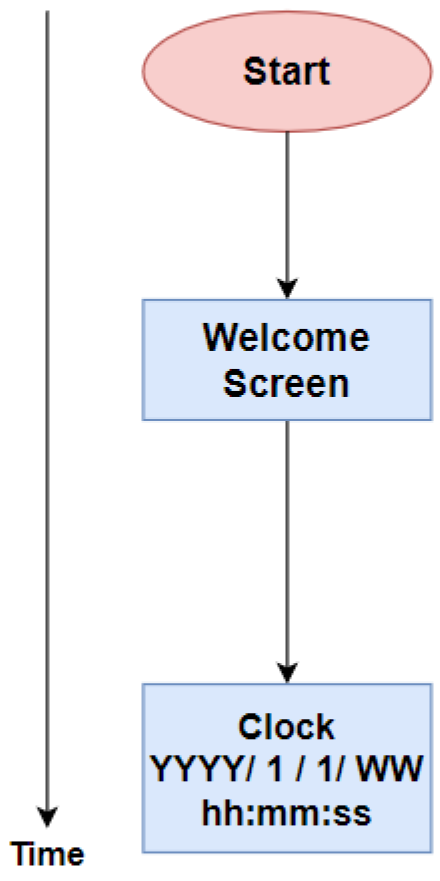
1. Normal time display
2. Alarm
3. Stopwatch

Our main priority is to achieve an independent cooperation of these three modes. For example, when a user decides to set the alarm or stopwatch, the time should work in a background mode without being delayed or reset.

Clock

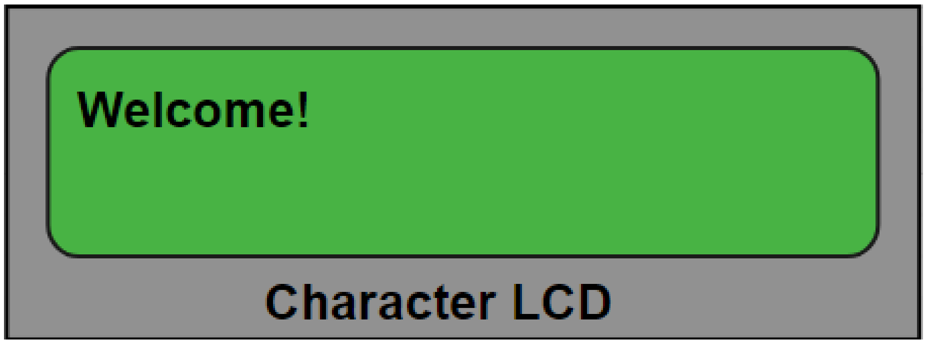
While working in a normal mode, digital clock should print the time, date (year, month, day) and day of the week.

Flow chart is shown below for a normal operation of a clock.



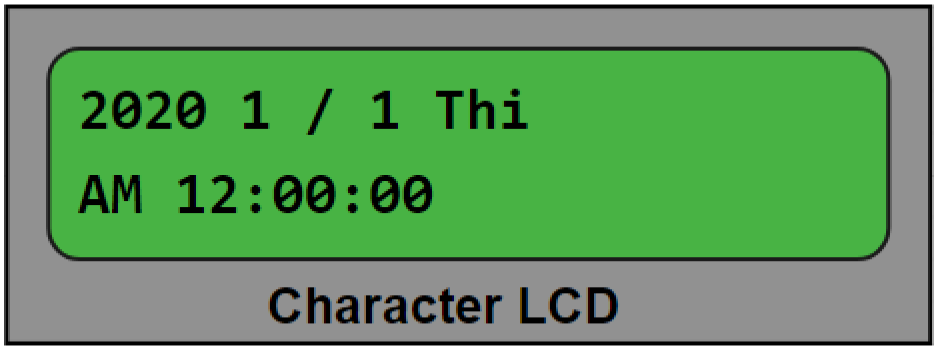
Expected output:

When the microcontroller is bootstrapped it will print a greeting message to a user.



```
Welcome!
```

A greeting message is displayed for two seconds after which the clock is being displayed by default.



```
2020 1 / 1 Thi
AM 12:00:00
```

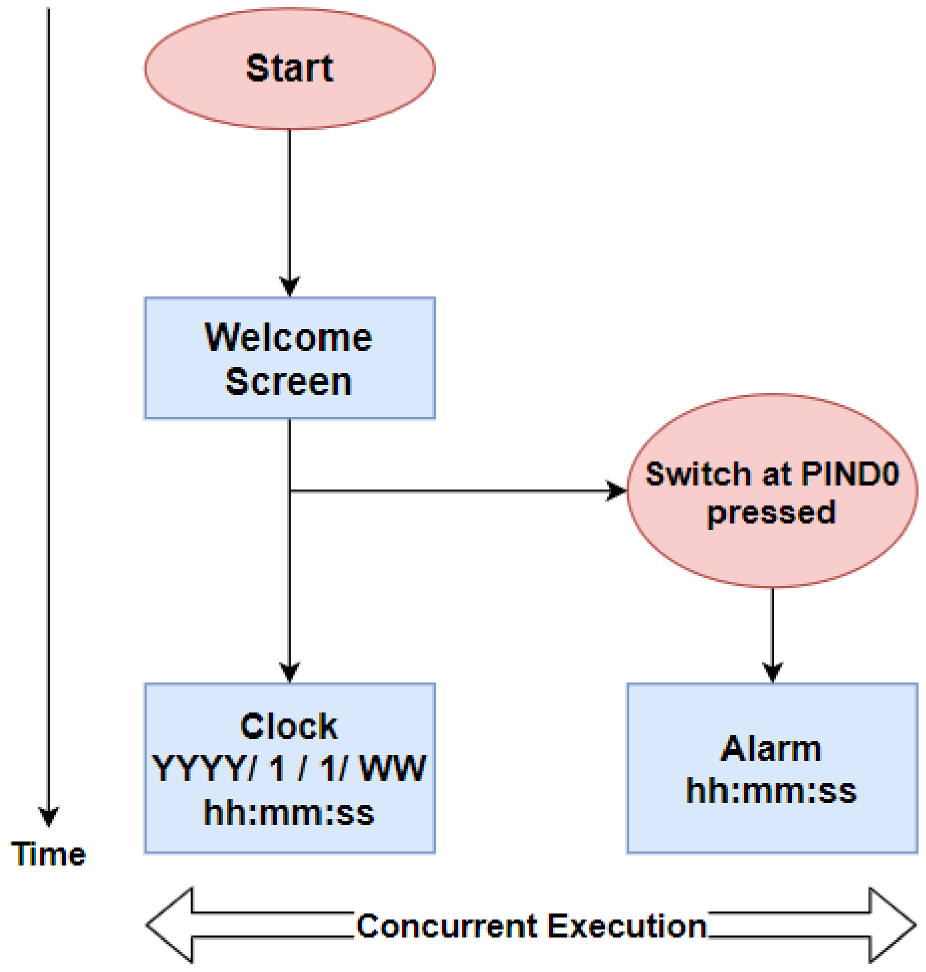
Alarm

Alarm - is a clock that is designed to alert an individual at a specified time. The primary function of these clocks is to awaken people from their night's sleep or short naps;

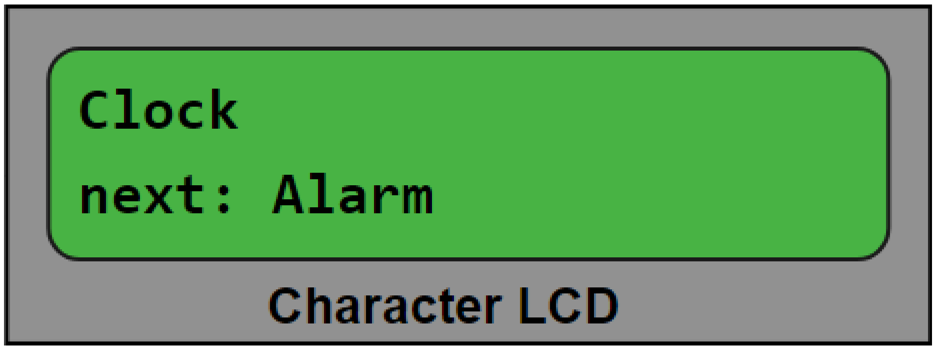
In order to switch between to alarm mode, a user should press a button connected to PIND0.

Important: Switching to the alarm mode shouldn't break up a normal operation of the clock. All the computations of the clock must be performed transparently and independently.

The flow chart below shows the working principle of alarm:



Expected output:



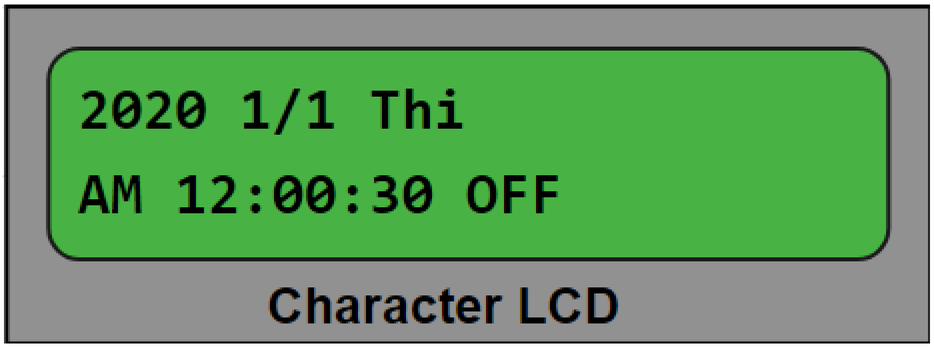
```
Clock
next: Alarm
```

By pressing a switch connected at PIND0, we change the mode from “Clock” to “Alarm”.

We can see alarm’s configuration screen.

To switch between any of the settings, a user has to press a button at PIND1.

By pressing PIND2, a user can jump change the value according the focus.

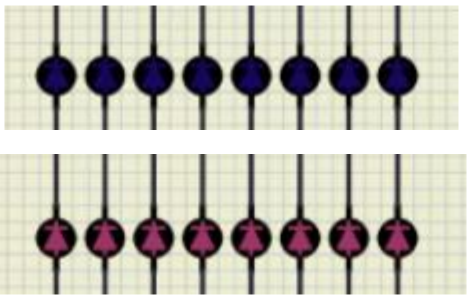


```
2020 1/1 Thi
AM 12:00:30 OFF
```

In this example, we’ve set the alarm for 12:00:30. When the clock comes ticks up to this time, several alarming effects will be activated.

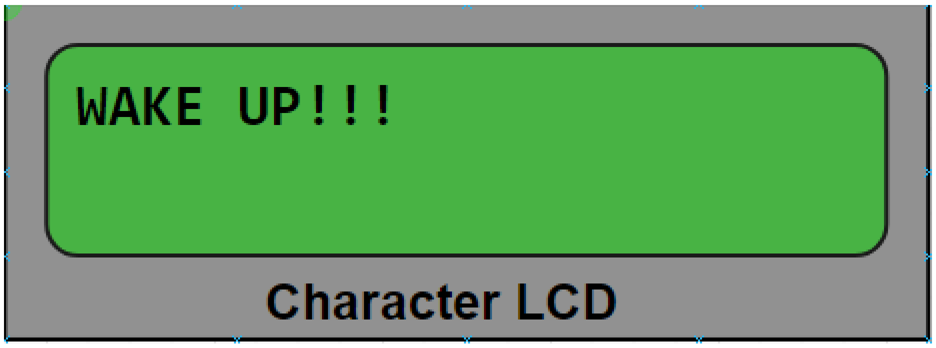
One of the possible effects is “Blinking LEDs”.

The pictures below show how a blinking should happen:



In addition to LEDs, the main GLCD screen will display a special text “WAKE UP!!!”.

The text can be seen on the picture below:



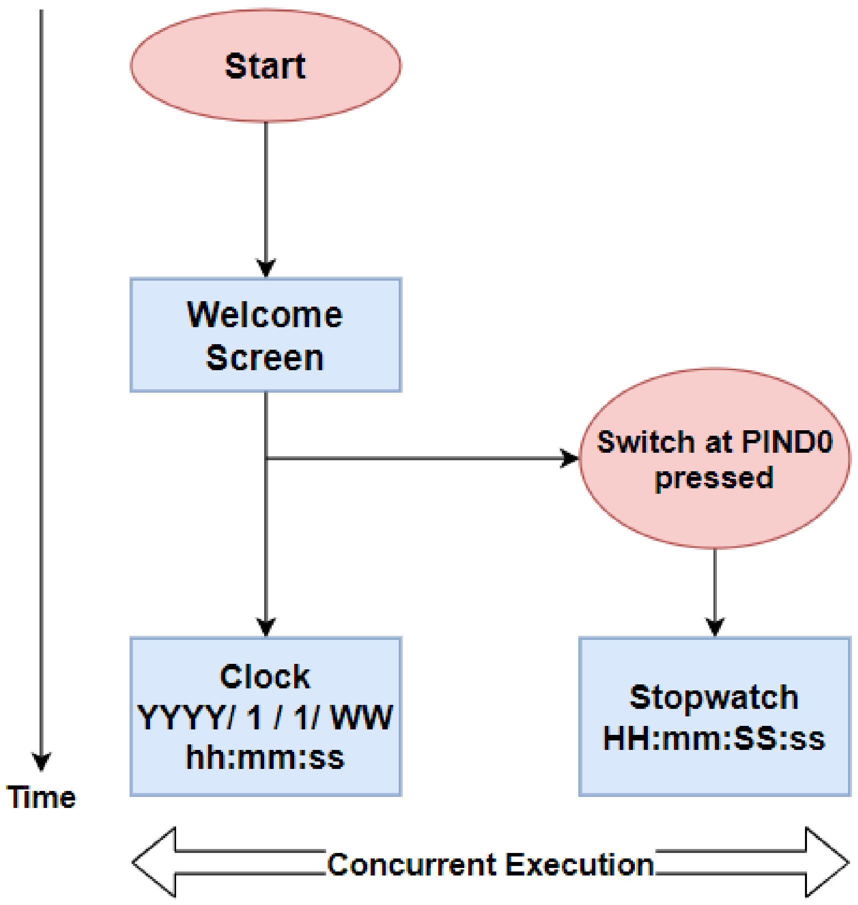
Stopwatch

Stopwatch is a handheld timepiece designed to measure the amount of time that elapses between its activation and deactivation. The clock is started and stopped by a person pressing a button. The timing functions are traditionally controlled by two buttons on the case. Pressing the top button starts the timer running, and pressing the button a second time stops it, leaving the elapsed time displayed.

In this project, the resolution of the stopwatch will be 1/100 of the sec.

Important: Switching to stopwatch mode shouldn’t break up a normal operation of the clock. All the computations of the clock must be performed transparently and independently.

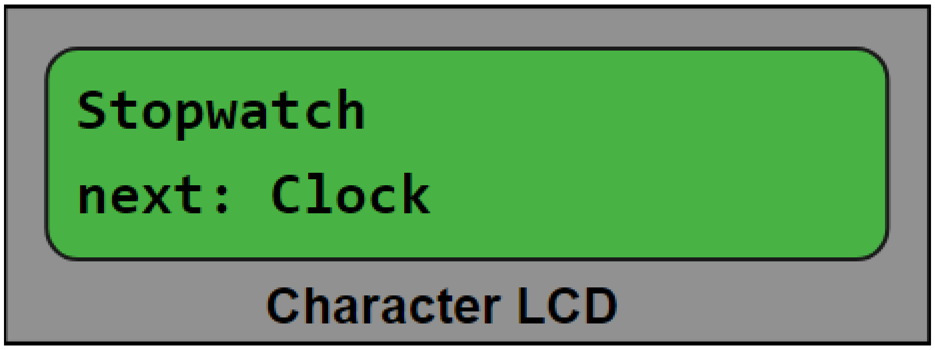
The flow chart below shows the working principle of stopwatch:



Expected output:

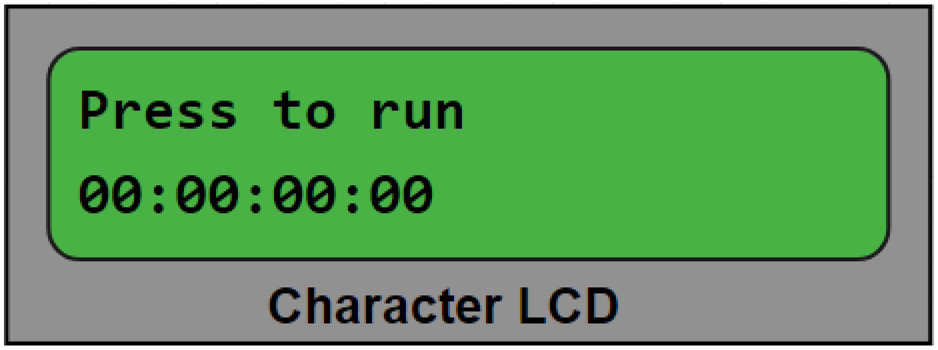
By pressing a switch connected at PIND0, we change the mode from “Clock” to “Stopwatch”.

After waiting for 1 second, we are able to see stopwatch’s configuration screen.



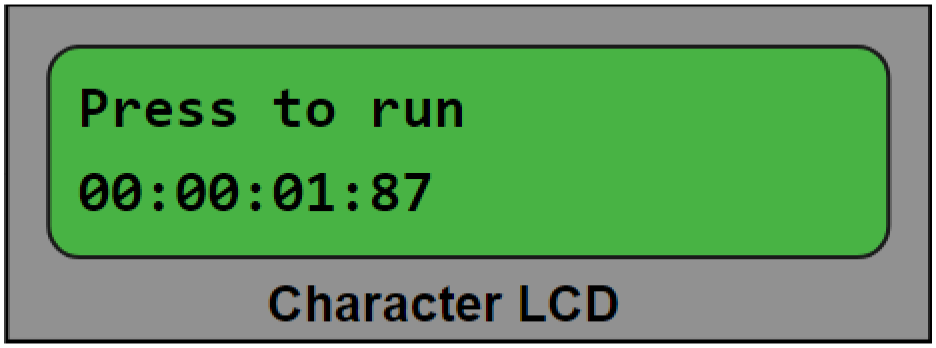
```
Stopwatch
next: Clock
```

By pressing a switch at PIND1 we will cause the stopwatch to run.



```
Press to run
00:00:00:00
```

The working process of the stopwatch can be seen in the picture below:



```
Press to run
00:00:01:87
```

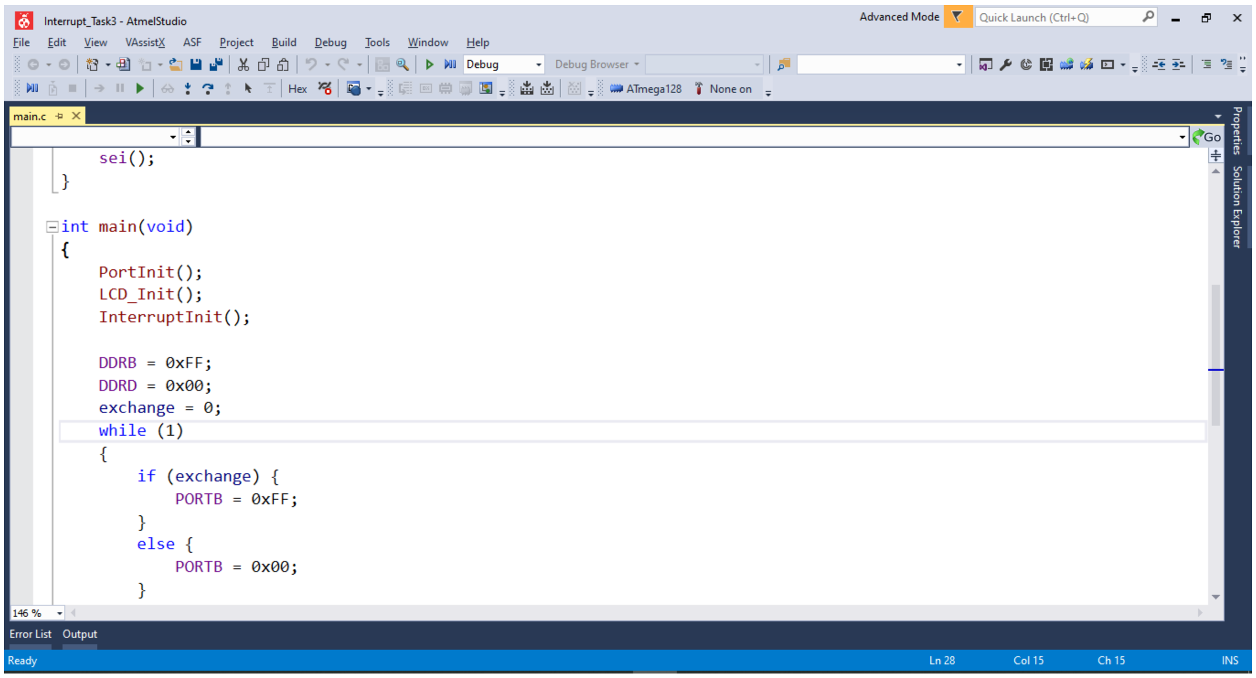
If we press again the switch at PIND1 we will cause the stopwatch to stop.

Software Requirements

AtmelStudio

AtmelStudio is the integrated development platform (IDP) for developing and debugging all AVR and SAM microcontroller applications. The Atmel Studio 7 IDP gives 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 the debuggers, programmers and development kits that support AVR and SAM devices.

Additionally, Studio includes Atmel Gallery, an online app store that allows you to extend your development environment with plug-ins developed by Microchip as well as third-party tool and embedded software vendors. Studio 7 can also seamlessly import your Arduino sketches as C++ projects, providing a simple transition path from Makerspace to Marketplace.

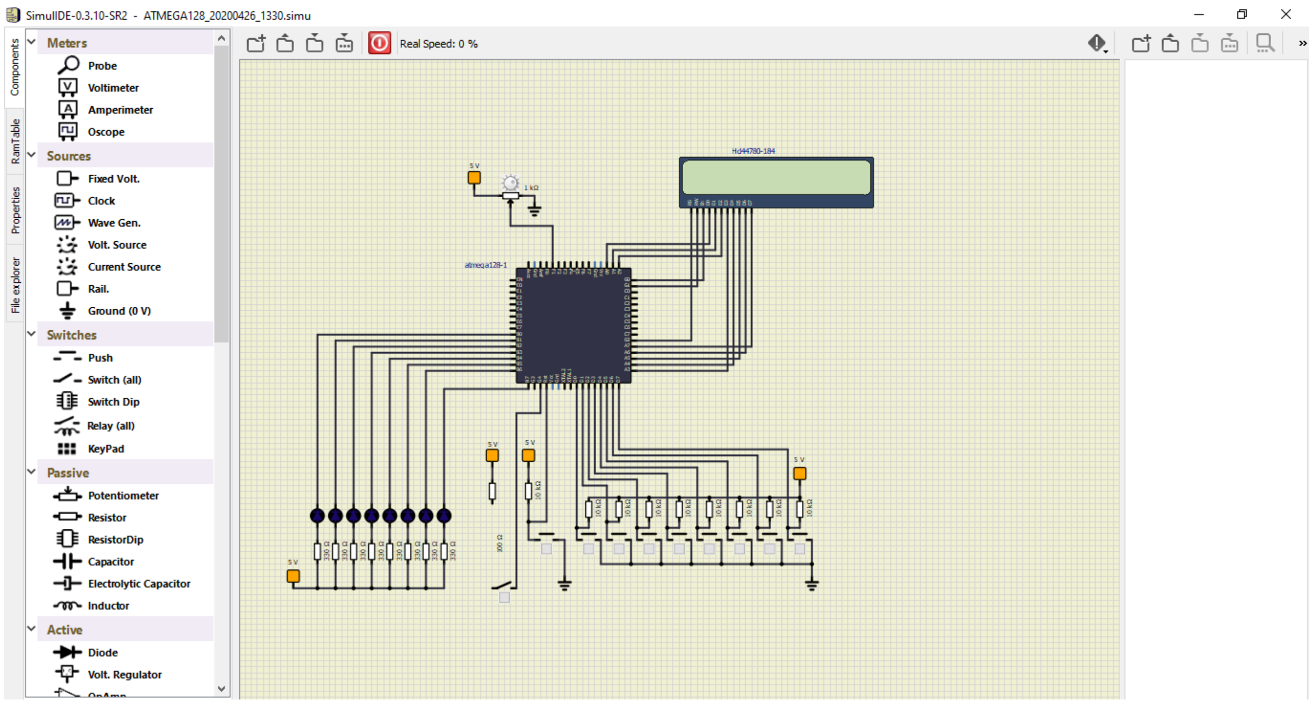


```
    sei();
}

int main(void)
{
    PortInit();
    LCD_Init();
    InterruptInit();

    DDRB = 0xFF;
    DDRD = 0x00;
    exchange = 0;
    while (1)
    {
        if (exchange) {
            PORTB = 0xFF;
        }
        else {
            PORTB = 0x00;
        }
```

SimulIDE

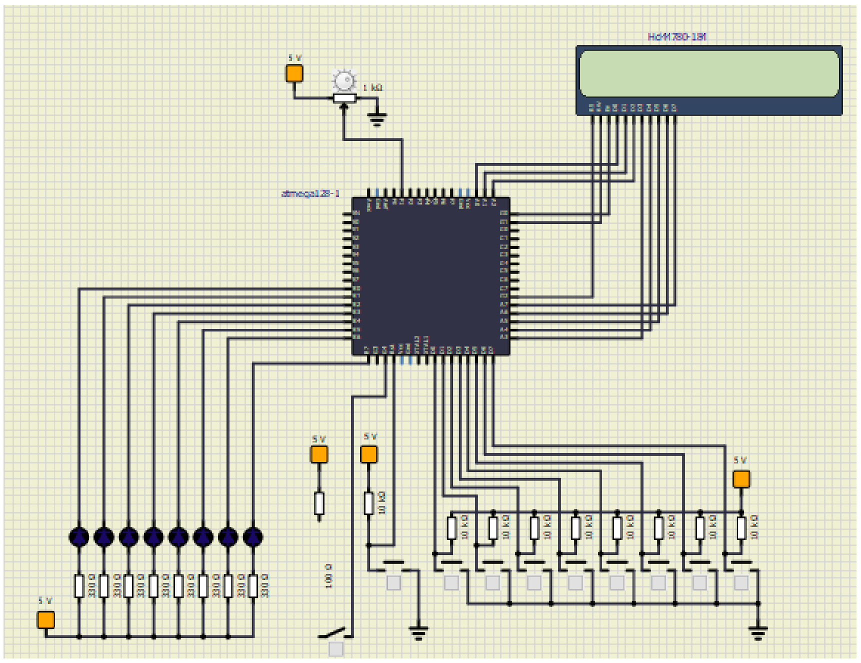
SimulIDE is a real-time electronic circuit simulator. It is a simple tool intended for advance learning, and it lets you enjoy the experience as well. SimulIDE is designed to be fast and easy to use, and it works wonders for simple electronics.

SimulIDE provides AVR, Arduino and PIC microcontrollers that can be accessed just like other components. Features like gpsim and simavr allow you to use PIC and AVR microcontrollers, respectively.



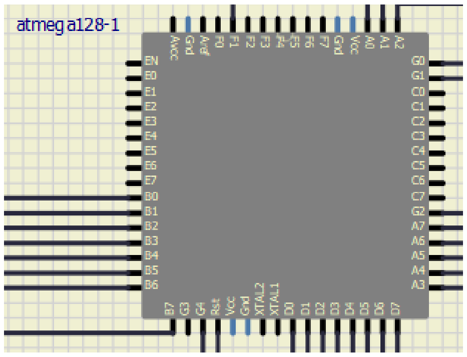
Circuit Requirements

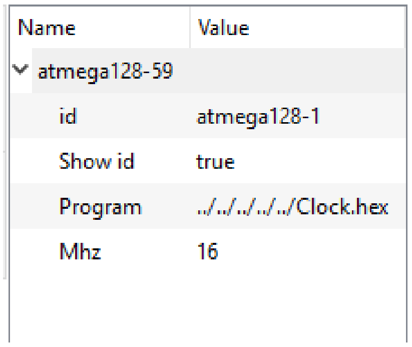
We can connect probes, LEDs, etc and view the output of our circuit according to our requirement. Buffers and logic gates like AND, OR and XOR are available to connect anywhere in the circuit. The readily-available AVR also need no prior setup, and can simply be added and connected directly in the circuit.



ATMega128

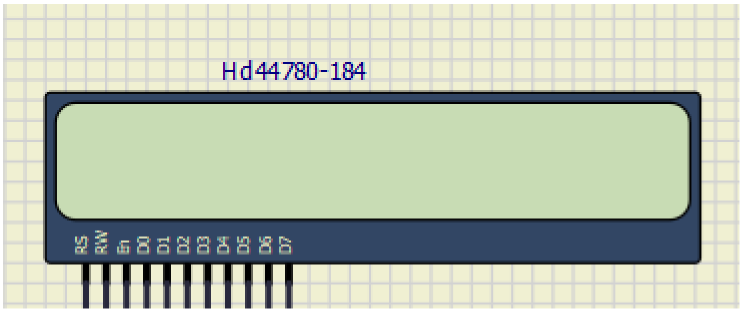
ATMega128 - high-performance, low-power AVR RISC-based microcontroller combines 128KB of programmable flash memory, 4KB SRAM, a 4KB EEPROM, an 8-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device supports throughput of 16 MIPS at 16 MHz and operates between 4.5-5.5 volts.

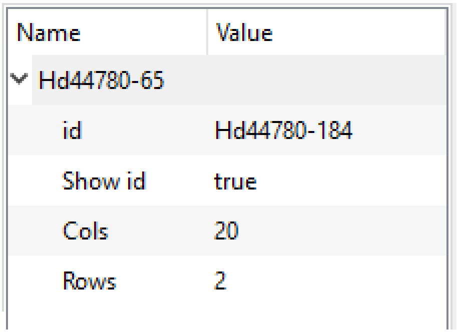

atmega128-1
AVcc Gnd Aref F0 F1 F2 F3 F4 F5 F6 F7 Gnd Vcc A0 A1 A2
E0 E1 E2 E3 E4 E5 E6 E7 B0 B1 B2 B3 B4 B5 B6
G0 G1 C0 C1 C2 C3 C4 C5 C6 C7 G2 A7 A6 A5 A4 A3
B7 G3 G4 Rst Vcc Gnd XTAL2 XTAL1 D0 D1 D2 D3 D4 D5 D6 D7



Name	Value
atmega128-59	
id	atmega128-1
Show id	true
Program	../../../../../Clock.hex
Mhz	16

HD44780 LCD controller is an alphanumeric dot matrix liquid crystal display (LCD) controller developed by Hitachi. The character set of the controller includes ASCII characters, Japanese Kana characters, and some symbols in two 28-character lines. Using an extension driver, the device can display up to 80 characters





Name	Value
Hd44780-65	
id	Hd44780-184
Show id	true
Cols	20
Rows	2

LED

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

In this project we are going to use an array of 8 LEDs.