



TOSHKENT SHAHRIDAGI INHA UNIVERSITETI  
INHA UNIVERSITY IN TASHKENT

# Embedded System

## Report

**Team name:**

CD Projekt Green

**Team Members:**

Oybek Khakimjanov	U1610179
Dilmurod Nasibullaev	U1610058
Anvar Abdulsatarov	U1610025

## **Overview**

- Description
- User Manual
- Algorithm explanation
- Video URL
- Team member contribution

## Description

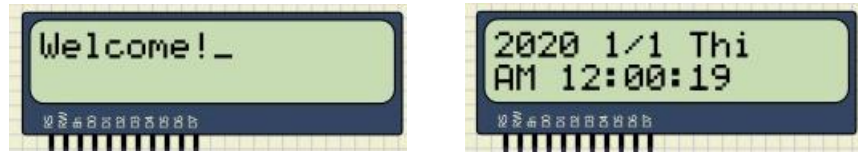
This document is designed to be a reference for any person wishing to find out about project or any person interested in the architecture and functionalities of **CD Projekt Green** project made on AVR microcontroller.

This document describes the general algorithm of project, provides video URL of demonstration of working project and presents team members' contribution to the project. Project is made on AVR microcontroller, standard kit specifically: AVR ATmega128 controller, Graphical LCD, board consisting of LEDs, switches, speaker. Project represents itself as Digital Alarm Clock with mode switching between **Clock**, **Alarm** and **Stopwatch**, different displaying methods as date and time, alarm with led blinking and stopwatch with resolution of 1/100 sec

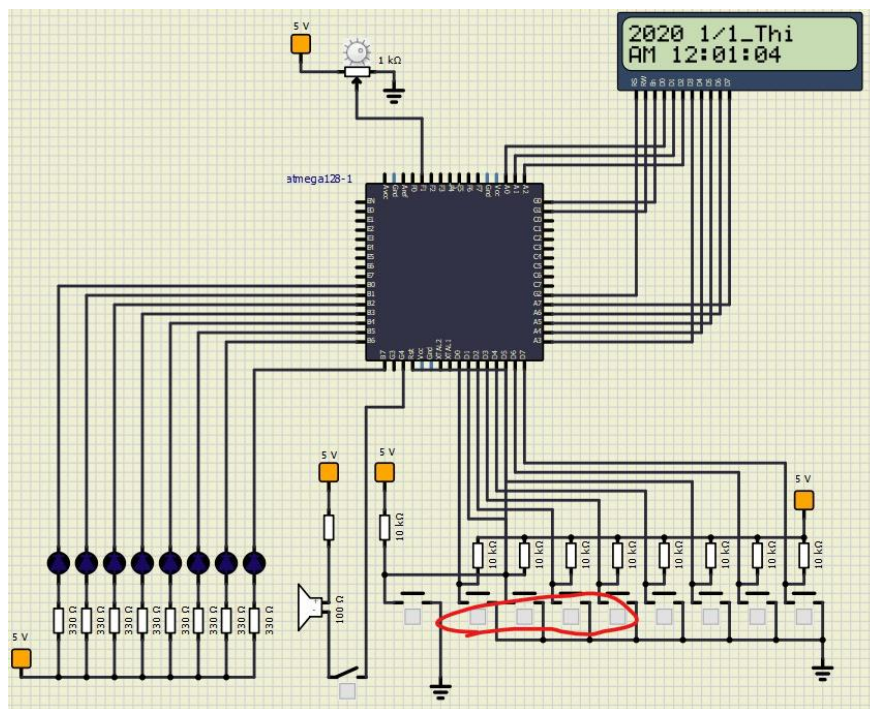
This document should be read by an individual with a technical background, who has experience in digital logic circuits, Assembly/C/Python languages, micro scheme designs, and development experience in object-oriented programming and event driven programming, LCD display's rendering principle.

# User Manual.

Turn on the watches. You will see the welcome screen and after that, you are ready to go.



There are 4 buttons:



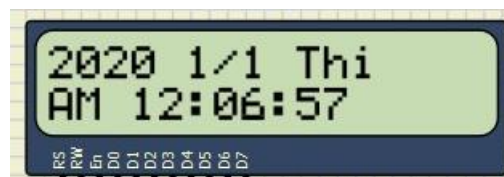
From left to right: *Button 0*, *Button 1*, *Button 2*, *Button 3*

There are three modes you can use:

- Clock mode – here you can know what actual time it is
- Alarm mode – you can set Alarm to any time
- Stopwatch mode – you can use Stopwatch with 1/100 sec precision

To change between modes, use *Button 0*

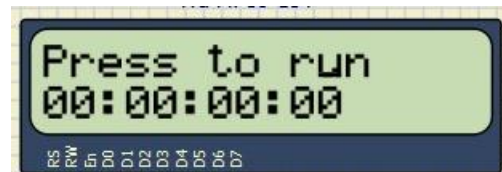
**Clock mode:**



**Alarm  
mode:**

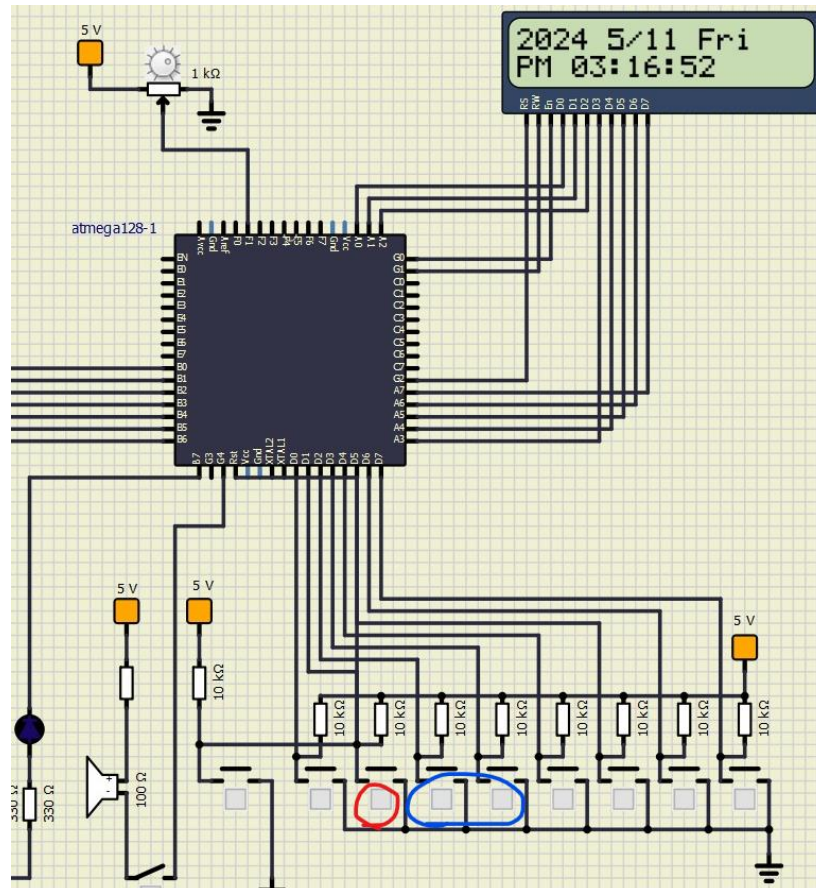


**Stopwatch  
mode:**



## Clock mode

In here use can change the value of anything you see on the screen by pressing **Button 1** to move the focus on what you want to change and then use **Button 2** or **Button 3** to increase or decrease the value respectively

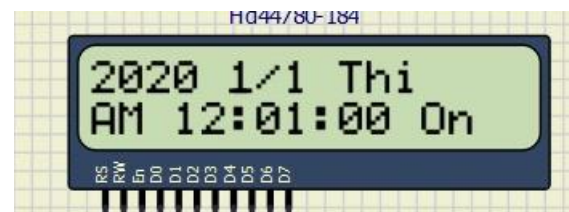


## Alarm mode.

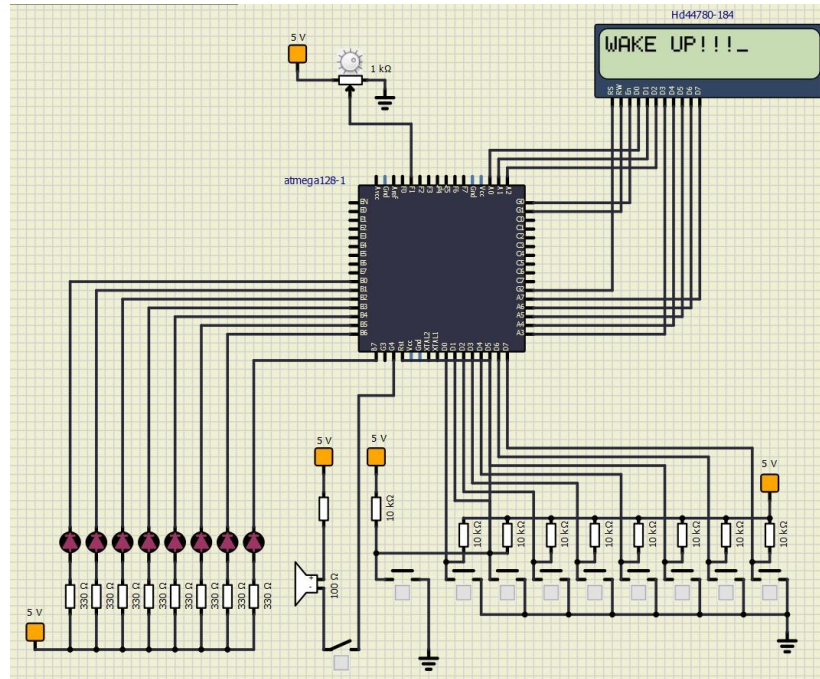
You do the same as in Clock mode. All buttons make the same function as there, but in Alarm mode you can also set the Alarm to **ON/OFF**.

If you want it to ring, just set the Time and Date in Alarm mode and then set Alarm to ON.

And then, when Time comes, the LED will blink with “WAKE UP!!!” sentence on the screen!







## Stopwatch mode

In Stopwatch mode, you can start stopwatch by pressing **Button 1** and stop it by pressing the same **Button 1**. It is easy as it is.

To reset Stopwatch, you can press **Button 2**



Figure 1 Stopwatch is Paused

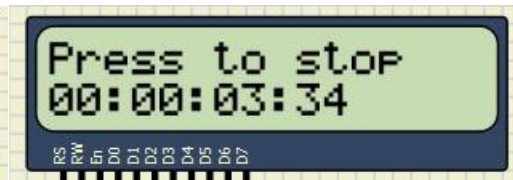


Figure 2 Stopwatch is Running



Figure 3 Stopwatch is Reset

### **Algorithm explanation.**

Everything related to Clock and Stopwatch values to be automatically changed, like **incrementing the second** is done in *Timer\_0*. We used **CTC mode** instead of normal mode to make it more precise.

In the beginning of the program, port, interrupt initialization and calculation of Leap year happen. Then welcomeScreen() function is called to display “Welcome!” on the screen.

The content of the screen is shown with the help of the function showScreen(), which is called inside main function’s while loop and it decides what content to show based on the value of mode global variable.

#### **Mode values:**

- **0: CLOCK**
- **1: ALARM**
- **2: STOPWATCH**

All the changes are done via interrupts. We used 4 interrupts in our program (0, 1, 2 and 3).

Interrupt\_0 serves for changing the value of the mode. So, if you press the button, the mode will be incremented. If it exceeds 2, then it is brought back to 0, as there are 3 modes.

#### **Clock mode:**

Initially, mode = 0, so that after “Welcome!”, the Clock is displayed on the screen and it instantly runs. You can change the values of the Clock any time, even if the clock is already running.

#### **In Clock mode (mode = 0), there are 8 variables displayed on the LCD:**

- Year
- Day
- Month
- Day of the week
- Part of Day (AM/PM)
- Hour
- Minute
- Second



To change the values of any of these variables, we need to use ***Interrupt\_1*** to first change a focus from one variable to the next one. If the focus is on the “Year” variable, then after ***Interrupt\_1***, the focus will be on “Day” variable.

To actually change the values, you need ***Interrupt\_2*** and ***Interrupt\_3***. They serve as “Arrow up” and “Arrow down” on the keyboard respectively. You can also say that they are like increment and decrement.

If you want to change specific variable, you need to focus on it by using ***Interrupt\_1***, then change the value by using ***Interrupt\_2*** or ***Interrupt\_3***.

## **Alarm mode**

Mode 1, and Alarm values do not change over time unless you change them by yourself, because it is NOT the Clock.

**In Alarm mode (mode = 1), there are 9 variables displayed on the LCD:**

- Year
- Day
- Month
- Day of the week
- Part of Day (AM/PM)
- Hour
- Minute
- Second
- Alarm On/Off

Controls are the same as in Clock mode, but in the end, you can also set **Alarm On/Off** to make it ring in specific time.

The check, whether the Time and Date of the Clock are the same as the Time and Date of the Alarm is made in Timer\_0 Interrupt. Every second it checks that and if they are the same AND the ***Alarm is ON***, then it makes global variable ***isAlarm = 1***.

The Alarm ring is specified in main function’s while loop.

Inside the loop, there is a check, whether ***isAlarm = 1 or 0*** and if it is 1, then the `invokeAlarm()` is called. And basically, what it does, it Shows “Wake up!!!” on the LCD screen 8 times along with LEDs are blinking 8 times. **IMPORTANT:** Clock

still works in the background even if the alarm is ringing. After the Alarm rang, the Clock, Alarm or Stopwatch (depending on the current mode) again is shown on the screen.

## Stopwatch mode

Mode 2.

**In Stopwatch mode (mode = 2), there are 4 variables displayed on the LCD:**

- Hours
- Minutes
- Seconds
- Milliseconds (1/100 of second in this case, but we called the value as millisecond to make it simpler to understand)

The logic is not the same as in previous modes.

***Interrupt\_1*** is to start and stop stopwatch

***Interrupt\_2*** is to reset all the values of **Stopwatch structure** to 0

And ***Interrupt\_3*** is not set at all, because there is no need for it to be usable, as stopwatch functionality is already satisfied by ***Interrupt 0, 1 and 2***

## Functions explanation.

***Brief Explanation of every major function. If you want detailed explanation, look at the source code. There are more detailed line-by-line comments***

### Initialize interrupts

```
void interruptInit(void){  
    // Falling edge for INT0, INT1, INT2, INT3  
    EICRA = 0xAA; // 1010 1010  
    EIMSK = 0x0F; // 0000 1111 enable INT0, INT1, INT2, INT3  
    DDRD = 0x00; // PORTD as input  
    sei(); // set Interrupt flag to 1  
}
```

## Start clock

```
void startTime(void){
    TCCR0 = 0x0f;    // CTC mode, Prescale 1024
    OCR0 = 99;       // count 0 to 99
    TIMSK = 0x02;    // Output compare interrupt enable 0000 00010
}
```

## Timer 0 interrupt

Formula to calculate condition in while loop to increment second

$TOV0 = (1 / (14.7456\text{Mhz}) * 1024(\text{prescale}) * 100 = 6.94444... \text{ ms}$

$6.94444... \text{ ms} * 144 = 1 \text{ sec}$

```
ISR(TIMER0_COMP_vect){ }
```

## Change the mode

```
SIGNAL(INT0_vect){ }
```

## Interrupt 1

Move cursor to right

```
SIGNAL(INT1_vect){ }
```

## Interrupt\_2

ArrowUp, increment the value, the cursor is pointing to

```
SIGNAL(INT2_vect){
    // Interrupt 2
    switch(mode){
        case 0:
            // watches
            break;
        case 1:
            // Alarm mode
            // the logic is the same except there is a case 8 to make isAlarmOn = 0 or 1
            break;
        case 2:
            // Resets stopwatch
            break;
        default:
            break;
    }
}
```

### **Interrupt\_3**

Arrow down, Decrement the value

Same logic as in In Interrupt\_2, but decrements values or makes them maximum when there is "borrow" like if minute is 0, then it is set to 59.

And also, there is no case for stopwatch as it does not need any other functionalities:

```
SIGNAL(INT3_vect){  
    switch(mode){  
        case 0:  
            // watches  
            break;  
        case 1:  
            // alarm  
            break;  
        default:  
            break;  
    }  
}
```

### **Devices initialization**

```
void initDevices(){ }
```

**"Welcome!" is shown for 2 seconds whenever the program is started**

```
void welcomeScreen(){ }
```

Basically, this function shows text on LCD screen in first and second lines respectively depending on the mode (Clock, Alarm, Stopwatch)

**This function is called in main() function's while loop**

```
void showScreen(){
    switch(mode){
        case 0:
            // If mode is changed (Interrupt_0 is pressed), then it shows title
of the
            // mode for 2 seconds
            break;
        case 1:
            break;
        case 2:
            break;
        default:
            break;
    }
}
```

**Makes LED blink 8 times and displays “WAKE UP!!!” on LCD**

```
void invokeAlarm(){ }
```

**Calculates whether it is leap year or not and then puts the value in daysInMonth array for February**

Is Leap year => daysInMonth[1] = 29

Is NOT Leap year => daysInMonth[1] = 28

```
void isLeapYear(){ }
```

**Main function with initialization, showScreen() and invokeAlarm() calls in while loop**

```
int main(void)
{
    DDRD = 0x00;           // PortD is Input
    DDRB = 0xFF;           // PortB is Output
    PORTB = 0xFF;          // Disable all LEDs
    initDevices();          // Initialize devices
    welcomeScreen();        // Initially show "Welcome!" for 2 seconds
    isLeapYear();           // Calculate whether it is leap year or not
    while(1){
        // Shows the content on the screen
        showScreen();
        if(isAlarm != 0){
            // If alarm is set and it should ring now, then it rings
            invokeAlarm();
        }
    }
}
```

## **Video URL**

<https://youtu.be/gtXSm5hVE8I>

## **Team members' roles**

**Dilmurod Nasibullaev** – as team leader, organized team meetings, gave us directions to think and shared his view on implementation of Final Project. Helped every team member with their tasks

**Anvar Abdulsatarov** – wrote the Code and provided ideas and solutions to the Final Project, participated in Recording the video. Made a research on how Timers work and how to make Timer more precise.

**Oybek Khakimjanov** – shared his ideas and solutions with the team. Focused our attention on most important parts of the tasks. Made a research regarding to the Atmega128 documentation and Timers.

**All team members participated in writing Report, Detail SW design specification document and Requirement analysis document**