Lab 7: Alexandre TRUONG

Link to this file in my GitHub repository:

https://github.com/AkaiRyussei/Digital-electronics-2

Analog-to-Digital Conversion

1. Complete table with voltage divider, calculated, and measured ADC values for all five push buttons.

Push button	PC0[A0] voltage	ADC value (calculated)	ADC value (measured)
Right	0 V	0	0
Up	0.495 V	101	99
Down	1.203V	246	257
Left	1.97V	403.062	409
Select	3.18V	650.628	639
none	5V	1023	1023

2. Code listing of ACD interrupt service routine for sending data to the LCD/UART and identification of the pressed button. Always use syntax highlighting and meaningful comments:

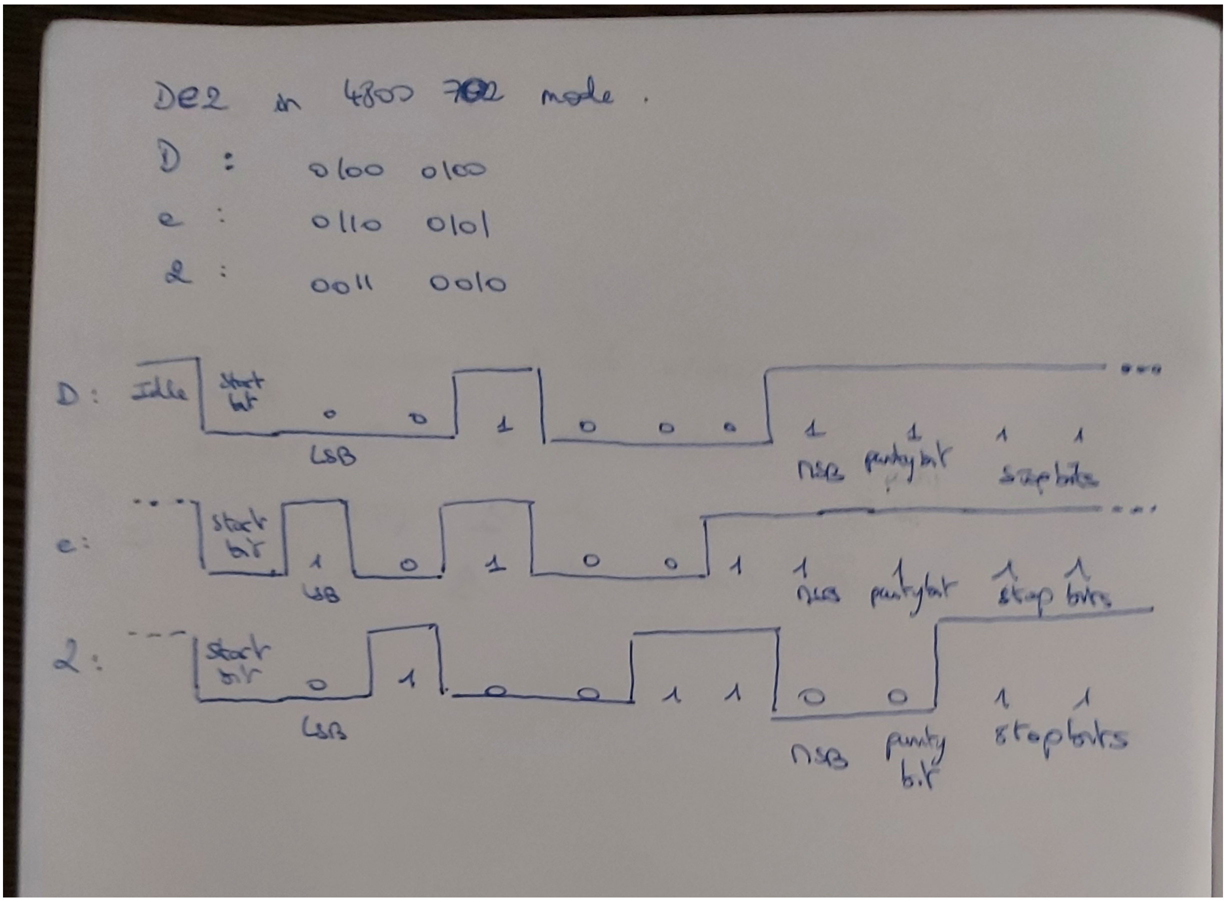
```c
/**********************************************************************
 * Function: ADC complete interrupt
 * Purpose:  Display value on LCD and send it to UART.
 **********************************************************************/
ISR(ADC_vect)
{
    uint16_t value = 0;
    char lcd_string[4] = "0000";

    value = ADC;                    // Copy ADC result to 16-bit variable
    itoa(value, lcd_string, 10);    // Convert decimal value to string

    // WRITE YOUR CODE HERE

    lcd_gotoxy(8, 0);
        lcd_puts("    ");
        lcd_gotoxy(8, 0);
        lcd_puts(lcd_string);

        //Send ADC value to UART Tx
        uart_puts(lcd_string);
        uart_puts(" ");
```

```
//Display ADC value in hexa at position "b"

itoa(value, lcd_string, 16);
lcd_gotoxy(13,0);
lcd_puts("   ");
lcd_gotoxy(13,0);
lcd_puts(lcd_string);


//Display what button was pressed at position "c"

// set 'c' according to ADC value
if (value==1022){
        //none
        lcd_gotoxy(8, 1);
        lcd_puts("     ");
        lcd_gotoxy(8, 1);
        lcd_puts("none");
}
if(value>=97 && value<=103){
        //up
        lcd_gotoxy(8, 1);
        lcd_puts("     ");
        lcd_gotoxy(8, 1);
        lcd_puts("up");

}
if(value>=400 && value<=405){

        //left
        lcd_gotoxy(8, 1);
        lcd_puts("     ");
        lcd_gotoxy(8, 1);
        lcd_puts("left");

}
if(value>=240 && value<=250){
        //down
        lcd_gotoxy(8, 1);
        lcd_puts("     ");
        lcd_gotoxy(8, 1);
        lcd_puts("down");

}
if(value>=647 && value<=653){
        //select
        lcd_gotoxy(8, 1);
        lcd_puts("     ");
        lcd_gotoxy(8, 1);
        lcd_puts("select");

}
if(value==0){
        //right
        lcd_gotoxy(8, 1);
```

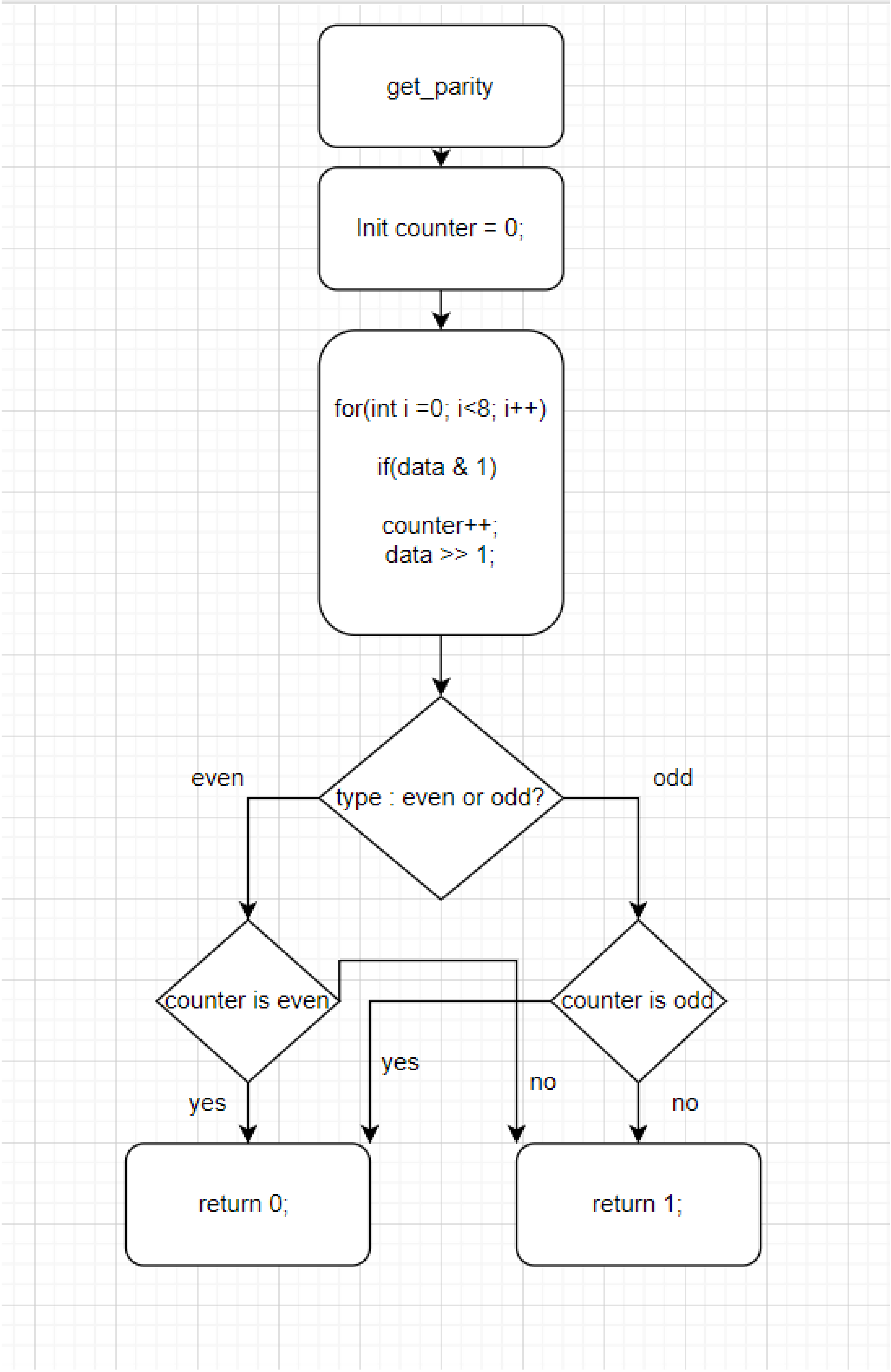
```
                lcd_puts("      ");
                lcd_gotoxy(8, 1);
                lcd_puts("right");
            }

        }
```

UART communication

1. (Hand-drawn) picture of UART signal when transmitting three character data `De2` in 4800 7O2 mode (7 data bits, odd parity, 2 stop bits, 4800 Bd).



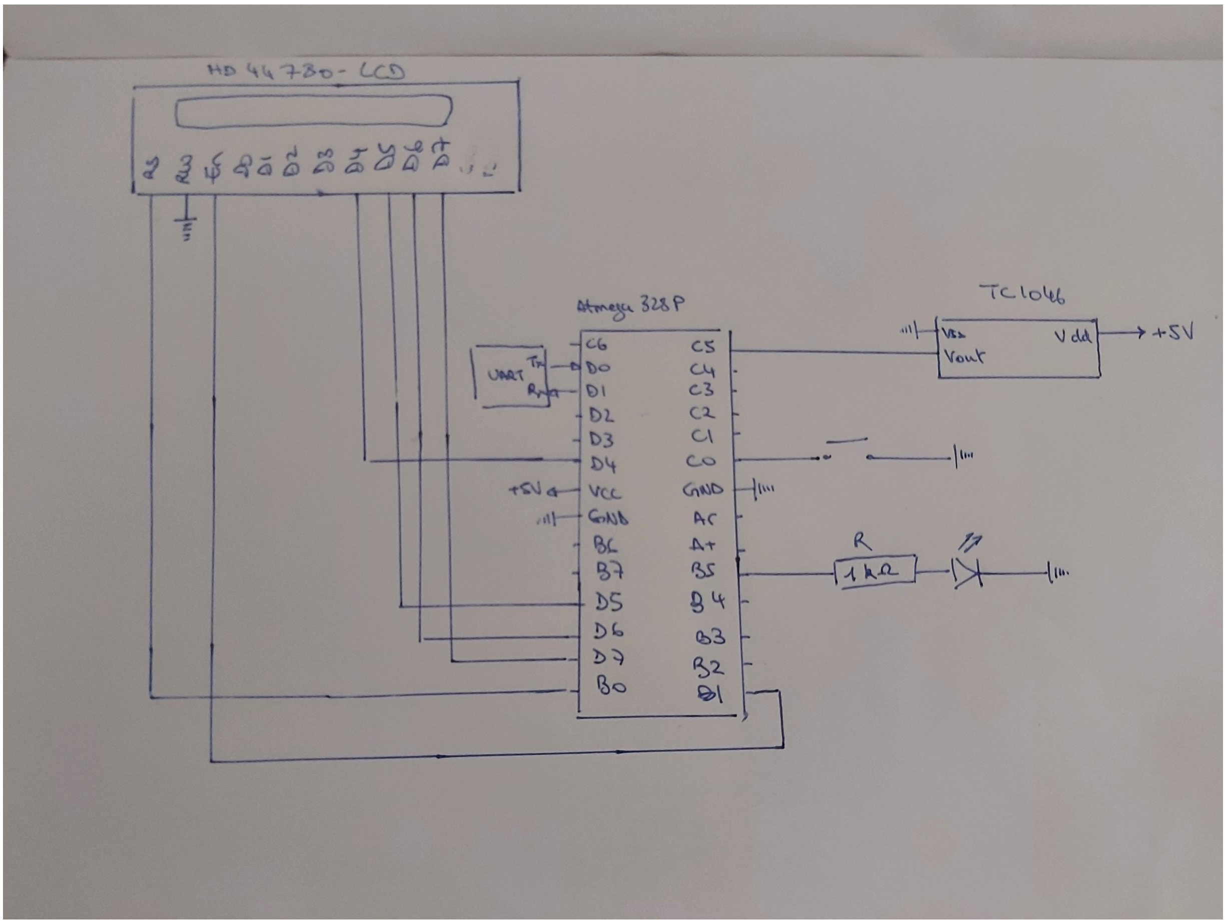
2. Flowchart figure for function `uint8_t get_parity(uint8_t data, uint8_t type)` which calculates a parity bit of input 8-bit `data` according to parameter `type`. The image can be drawn on a computer or by hand. Use clear descriptions of the individual steps of the algorithms.



Temperature meter

Consider an application for temperature measurement and display. Use temperature sensor TC1046, LCD, one LED and a push button. After pressing the button, the temperature is measured, its value is displayed on the LCD and data is sent to the UART. When the temperature is too high, the LED will start blinking.

1. Scheme of temperature meter. The image can be drawn on a computer or by hand. Always name all components and their values.


HD 44780 - LCD
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
Atmega 328P
C6
D0
D1
D2
D3
D4
VCC
GND
B6
B7
D5
D6
D7
B0
C5
C4
C3
C2
C1
C0
GND
Ar
A+
B5
B4
B3
B2
B1
UART
Tx
Rx
+5V
TC1046
Vss
Vout
Vdd
+5V
R
1kΩ