

 **xkocum00** Update README.md Latest commit a341ff7 1 hour ago 🕒 History

👤 1 contributor

☰ 112 lines (76 sloc) | 4.88 KB

<> 📄 Raw Blame 🖨 📄 ✎ 🗑

Lab 7: FILIP_KOCUM

Link to this file in your GitHub repository:

[github link of my repository](#)

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

Push button	PC0[A0] voltage	ADC value (calculated)	ADC value (measured)
Up	0.495 V	101	99
Down	1.203 V	246	256
Left	1.970 V	403	409
Select	3.182 V	651	639
none	5 V	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:

```

/*****
* Function: ADC complete interrupt
* Purpose: Display value on LCD and send it to UART.
*****/
ISR(ADC_vect)
{
    // WRITE YOUR CODE HERE

    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

    lcd_clrscr();
    lcd_gotoxy(1, 0); lcd_puts("value:");
    lcd_gotoxy(3, 1); lcd_puts("key:");

    if(value >= 1022){

```

```

lcd_gotoxy(8, 0); lcd_puts(lcd_string);
itoa(value, lcd_string, 16);
lcd_gotoxy(13,0); lcd_puts(lcd_string);
lcd_gotoxy(8, 1); lcd_puts("none  ");
uart_puts(lcd_string); uart_puts("\n\r");

```

```

}else if(value >= 630){

```

```

lcd_gotoxy(8, 0); lcd_puts(lcd_string);
itoa(value, lcd_string, 16);
lcd_gotoxy(13,0); lcd_puts(lcd_string);
lcd_gotoxy(8, 1); lcd_puts("Select ");
uart_puts(lcd_string); uart_puts("\n\r");

```

```

}else if(value >= 400){

```

```

lcd_gotoxy(8, 0); lcd_puts(lcd_string);
itoa(value, lcd_string, 16);
lcd_gotoxy(13,0); lcd_puts(lcd_string);
lcd_gotoxy(8, 1); lcd_puts("Left  ");
uart_puts(lcd_string); uart_puts("\n\r");

```

```

}else if(value >= 246){

```

```

lcd_gotoxy(8, 0); lcd_puts(lcd_string);
itoa(value, lcd_string, 16);
lcd_gotoxy(13,0); lcd_puts(lcd_string);
lcd_gotoxy(8, 1); lcd_puts("Down  ");
uart_puts(lcd_string); uart_puts("\n\r");

```

```

}else if(value >= 98){

```

```

lcd_gotoxy(8, 0); lcd_puts(lcd_string);
itoa(value, lcd_string, 16);
lcd_gotoxy(13,0); lcd_puts(lcd_string);
lcd_gotoxy(8, 1); lcd_puts("Up    ");

```

```

// Put ADC value in decimal

```

```

// Put ADC value in hexadecimal

```

```

// Put button name here

```

```

// Put ADC value in decimal

```

```

// Put ADC value in hexadecimal

```

```

// Put button name here

```

```

// Put ADC value in decimal

```

```

// Put ADC value in hexadecimal

```

```

// Put button name here

```

```

// Put ADC value in decimal

```

```

// Put ADC value in hexadecimal

```

```

// Put button name here

```

```

// Put ADC value in decimal

```

```

// Put ADC value in hexadecimal

```

```

// Put button name here

```

```

uart_puts(lcd_string); uart_puts("\n\r");

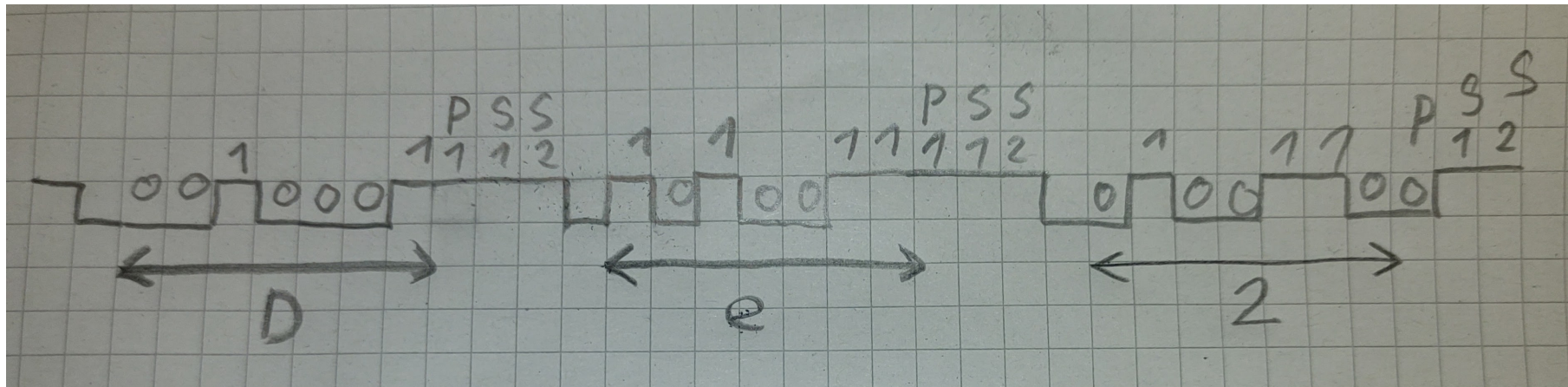
}else{

    lcd_gotoxy(8, 0); lcd_puts(lcd_string);           // Put ADC value in decimal
    itoa(value, lcd_string, 16);
    lcd_gotoxy(13,0); lcd_puts(lcd_string);           // Put ADC value in hexadecimal
    lcd_gotoxy(8, 1); lcd_puts("Right      ");        // Put button name here
    uart_puts(lcd_string); uart_puts("\n\r");
}

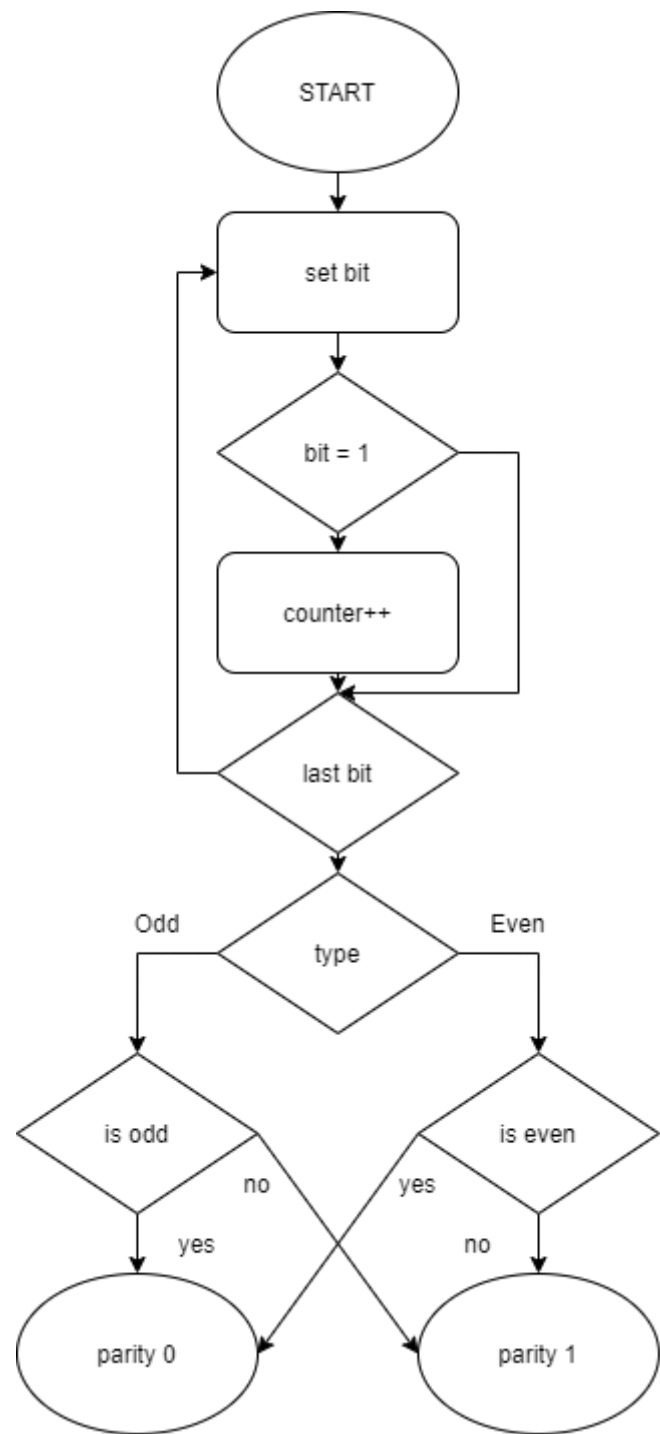
```

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.

