Lab 7: Pavel Vaněk

Link to my `Digital-electronics-2` GitHub repository:

https://github.com/Bobik77/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	1.203 V	246	257
Left	1.969 V	403	409
Select	3.18 V	651	639
none	5 V	1023	1023

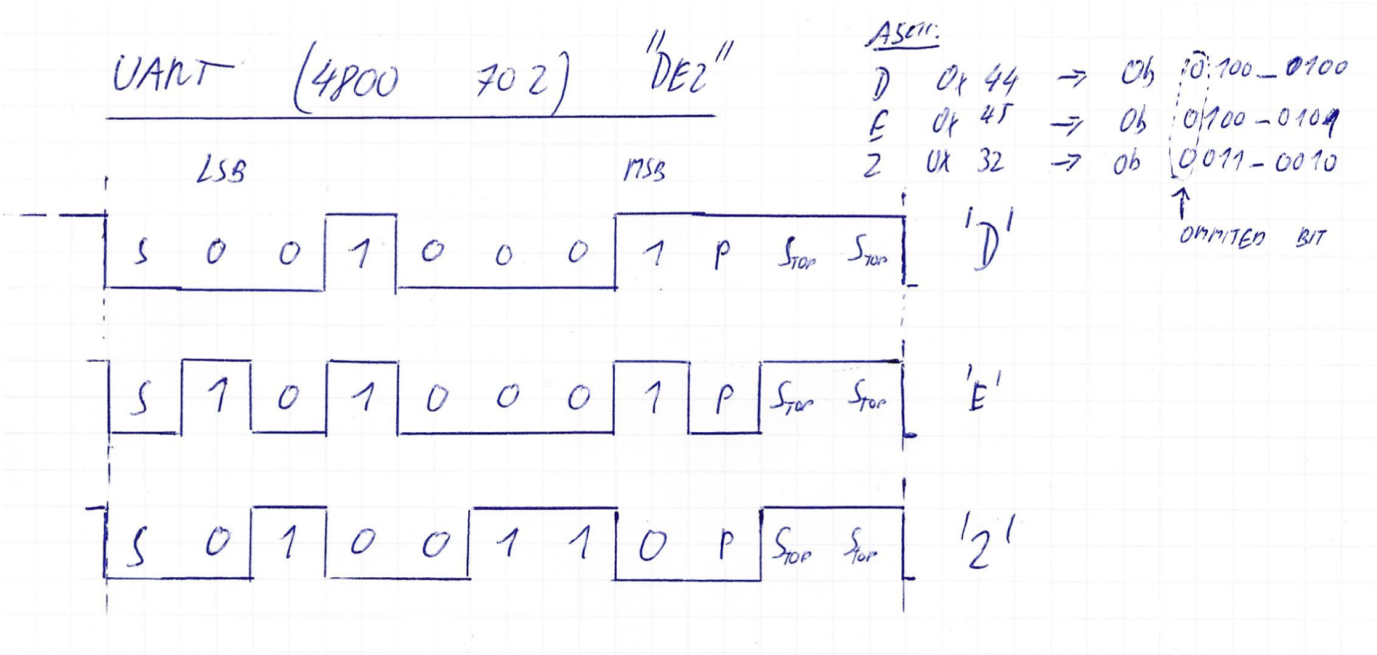
2. Code listing of ACD interrupt service routine for sending data to the LCD/UART and identification of the pressed button. Always use syntax highlighting and meaningful comments:

```
/* Constants -------------------------------------------------------------*/
//array of strings key names
const char *key_name_str[6] = { "NONE", "SELECT", "LEFT",
                                "DOWN","UP","RIGHT"};
```

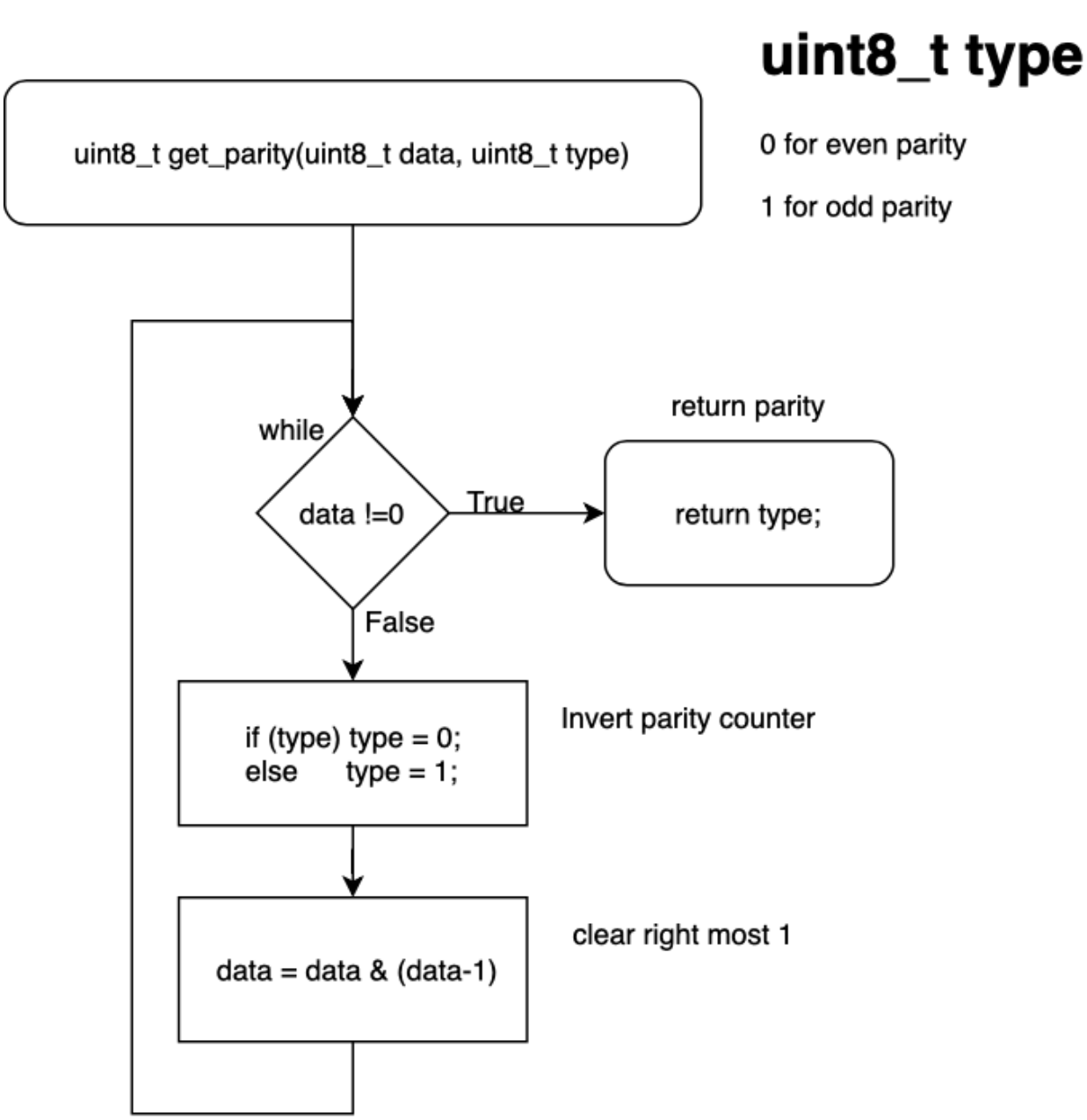
```
/*****************************************************************************
 * Function: ADC complete interrupt
 * Purpose:  Display value on LCD and send it to UART.
 *****************************************************************************/
ISR(ADC_vect)
{
    //clear first line
    lcd_gotoxy(8,0); //set cursor
    lcd_puts("        ");

    // read adc value
    uint16_t ADC_val = ADC;

    char lcd_string[4] = "    ";      // String for converting numbers by itoa()
    //show in DEC
    itoa(ADC_val, lcd_string, 10);     // Convert to str
    lcd_gotoxy(8,0); //set cursor to A
    lcd_puts(lcd_string);
    // show in HEX
    itoa(ADC_val, lcd_string, 16);     // Convert to str
    lcd_gotoxy(13,0); //set cursor to B
    lcd_puts(lcd_string);

    // Show addicted key-label
    uint8_t key = 0;
    if (ADC_val > 1020) key = 0; //NON
    else if (ADC_val > 600) key = 1; //SEL
    else if (ADC_val > 390) key = 2; //LEFT
    else if (ADC_val > 200) key = 3; //DWN
    else if (ADC_val > 50)  key = 4; //UP
    else key = 5; //RIGHT
    lcd_gotoxy(8,1); //set cursor to C
    lcd_puts("        "); //clear 2nd line disp
    lcd_gotoxy(8,1); //set cursor to C
    lcd_puts(key_name_str[key]); //show on LCD
    uart_puts(key_name_str[key]);//send via COM
    uart_puts("\n"); //line feed
}
```

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).



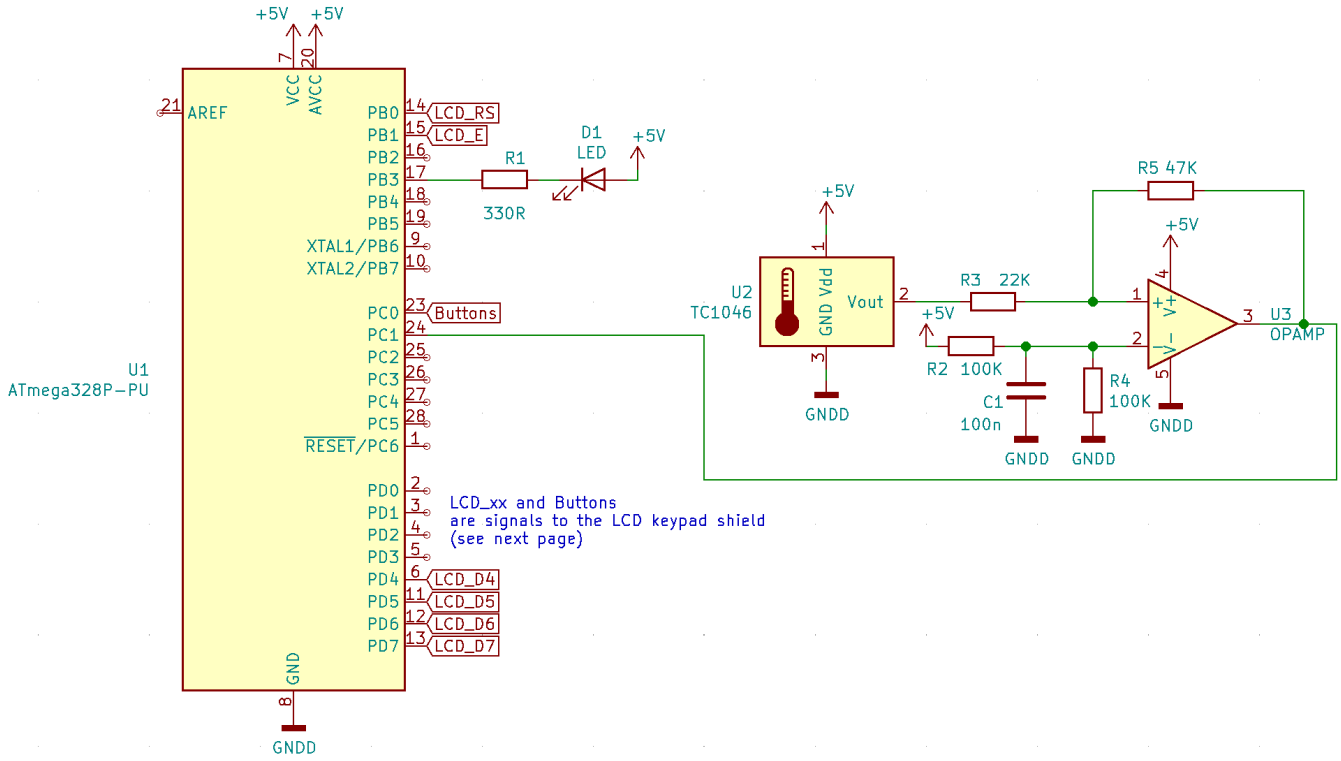
2. Flowchart figure for function `uint8_t get_parity(uint8_t data, uint8_t type)` which calculates a parity bit of input 8-bit `data` according to parameter `type`. The image can be drawn on a computer or by hand. Use clear descriptions of the individual steps of the algorithms.

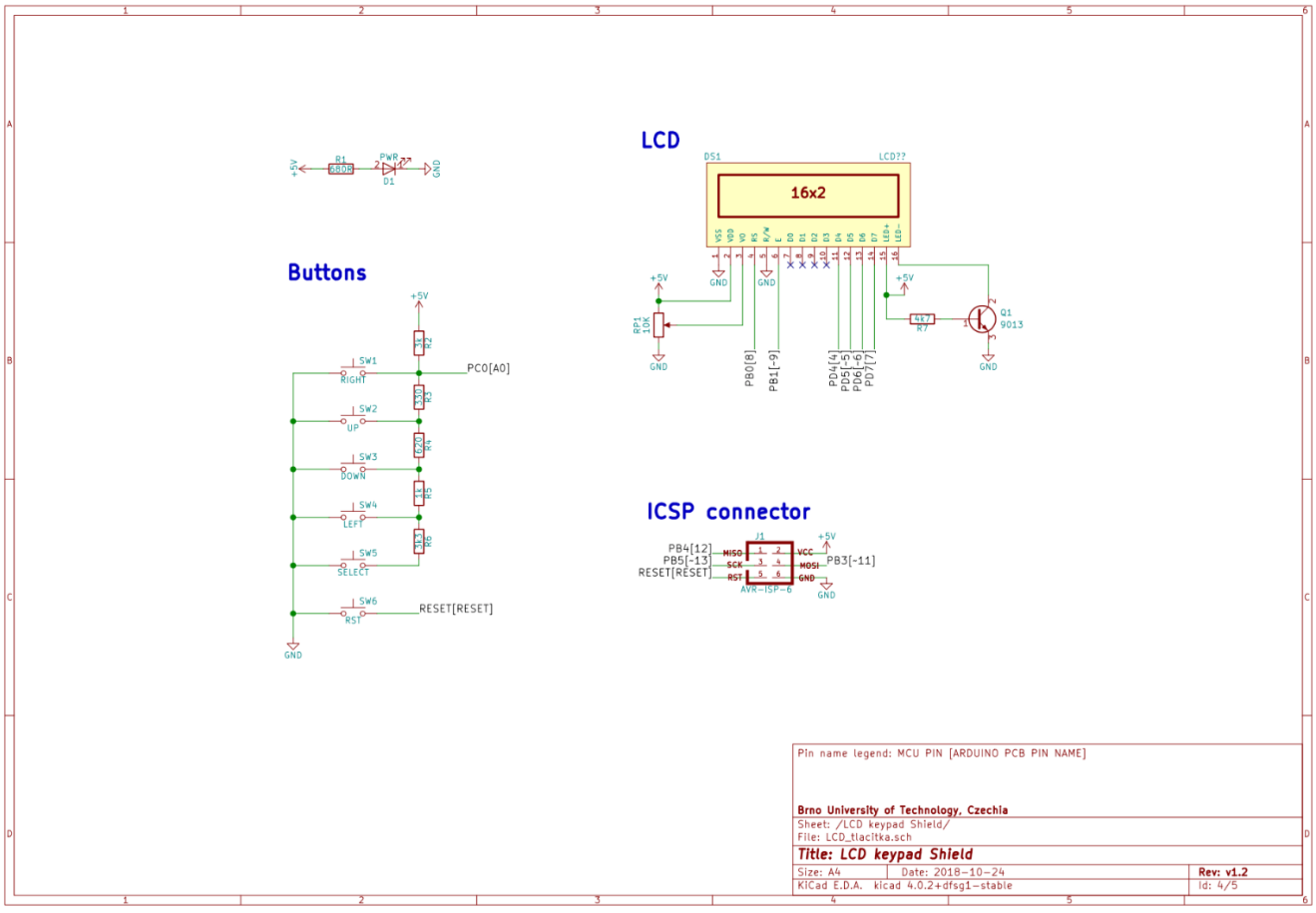


Temperature meter

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.


+5V
+5V
VCC
AVCC
AREF
PB0
PB1
PB2
PB3
PB4
PB5
XTAL1/PB6
XTAL2/PB7
PC0
PC1
PC2
PC3
PC4
PC5
RESET/PC6
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
GND
LCD_RS
LCD_E
LCD_D4
LCD_D5
LCD_D6
LCD_D7
Buttons
R1
330R
D1
LED
U1
ATmega328P-PU
GNDD
U2
TC1046
GND Vdd
Vout
R3 22K
R2 100K
C1
100n
R5 47K
R4
100K
U3
OPAMP
LCD_xx and Buttons
are signals to the LCD keypad shield
(see next page)


LCD
16x2
Buttons
SW1
RIGHT
SW2
UP
SW3
DOWN
SW4
LEFT
SW5
SELECT
SW6
RST
PC0[A0]
RESET[RESET]
ICSP connector
PB4[12]
PB5[-13]
RESET[RESET]
PB3[-11]
MISO
SCK
RST
VCC
MOSI
GND
AVR-ISP-6
Pin name legend: MCU PIN [ARDUINO PCB PIN NAME]
Brno University of Technology, Czechia
Sheet: /LCD keypad Shield/
File: LCD_tlacitka.sch
Title: LCD keypad Shield
Size: A4
Date: 2018-10-24
Rev: v1.2
KiCad E.D.A. kicad 4.0.2+dfsg1-stable
Id: 4/5