

Microcontroller 2 Logbook

Assignment 1: UART

Richie Valentino Richard Ilot

470819

ETI2V.IA

1. Time management

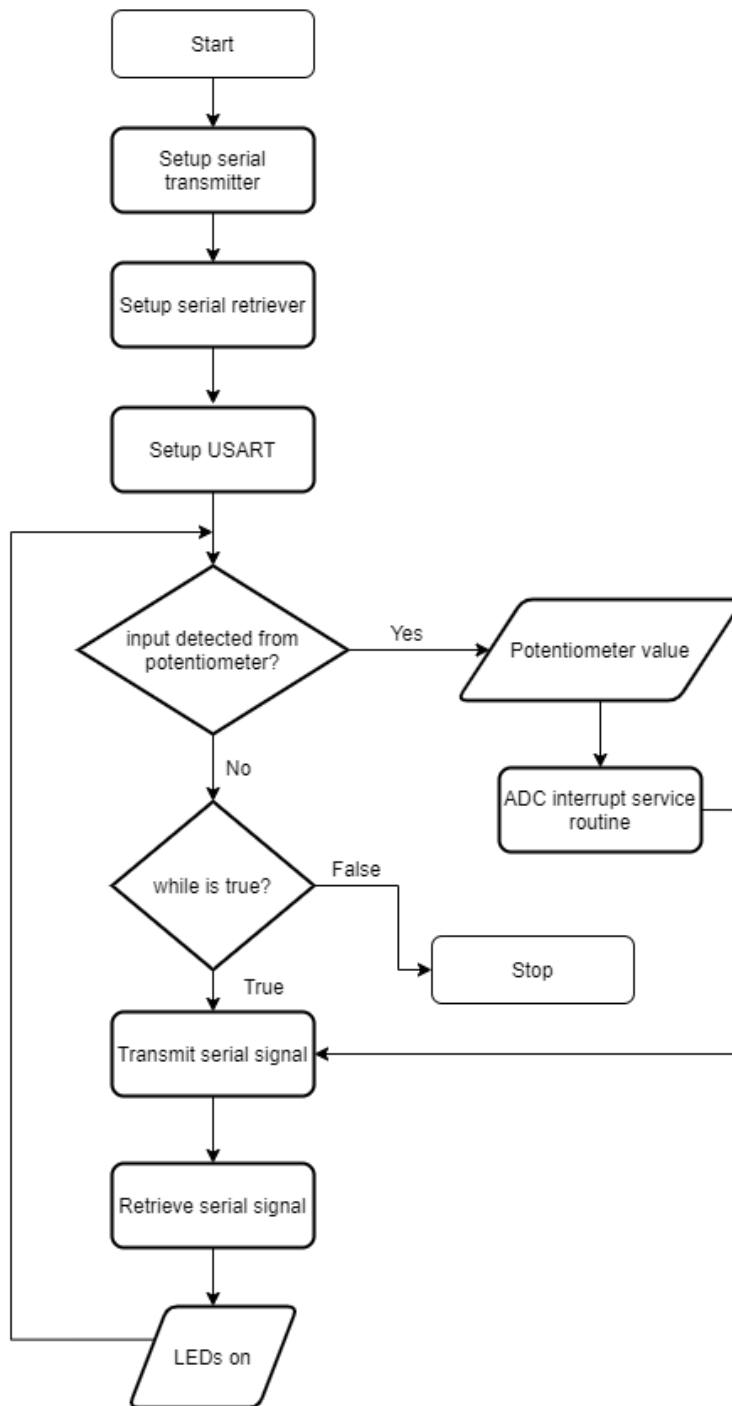
Total time spent for the preparation, lesson, and evaluation of this week is approximately 7 hours, approximately 2 hours are time spent in the lesson (e.g. preview, class lesson, review). The rest of the time, approximately 5 hours are time spent in the assignment.

2. Assignment 1: UART

Objectives of this assignment are:

- Connect 2 Arduinos with serial communication.
- Connect a potentiometer to each of the Arduino's analog input.
- Connect 4 LEDs to each Arduino.

2.1. Program flow diagram



2.2. Code snippets

```
/*
 * serial_comm_transmit.c
 *
 * Created: 09/02/2021 14:59:43
 * Author : Richie Iltot
 */

#define F_CPU 1600000UL
#define FOSC 1843200 //clk speed
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
#include <avr/io.h>
#include <avr/interrupt.h>

/**
 * Note to self: this program contains both transmitter and retriever.
 * Reason is to make the program more mobile to become transmitter/retriever.
 */

//USART setup function
void USART_setup(void){
    //set baud rate
    UBRR0H = (unsigned char) (MYUBRR >> 8);
    UBRR0L = (unsigned char) MYUBRR;

    //enable transmit (change RXEN0 to 1 if retrieve is enabled)
    UCSR0B = (1<<TXEN0)|(0<<RXEN0);

    //set 8 bit char and 2 stop bit
    UCSR0C = (1<<USBS0)|(3<<UCSZ00);
}

//USART retriever function
unsigned char USART_retrieve(void){
    while(!(UCSR0A & (1<<RXC0))){
        return UDR0;
    }
}

//USART transmitter function
void USART_transmit(unsigned char data){
    while( !(UCSR0A & (1<<UDRE0)) ){
        UDR0 = data;
    }
}
```

```

//pin and ADC setup for transmitter
void transmitter_setup(void){
    DDRC = 0b00000000; //sets all C pins as input
    /*
    ADMUX is ADC Multiplexer Selection Register.
    Setting bit 6 (REFS0) to high will select the
    voltage reference for ADC to AVcc (Table 23-3 in datasheet).
    And setting bit 5 (ADLAR) to '1' will adjust the result to the left.
    */
    ADMUX |= (1<<REFS0)|(1<<ADLAR);
    /*
    ADCSRA is ADC Control and Status Register A.
    Setting bit 7 (ADEN - ADC Enable) to high will enables the AD conversion
    Setting bit 6 (ADSC - ADC start conversion) to high will perform the AD conversion.
    Setting bit 5 (ADATE - ADC Auto Trigger Enable) to high will enable the auto triggering of the ADC.
    Setting bit 3 (ADIE - ADC Interrupt Enable) to high will enable interrupt for ADC is activated.
    Setting bit 2-0 (ADPS0-2) to high will select the ADC division factor for the prescaler to 128.
    */
    ADCSRA |= (1<<ADEN)|(1<<ADSC)|(1<<ADATE)|(1<<ADIE)|(1<<ADPS0)|(1<<ADPS1)|(1<<ADPS2);
}

void retriever_setup(void){
    DDRB = 0b00001111; //sets B pin 0-3 to output
}

int main(void)
{
    cli();
    transmitter_setup();
    retriever_setup();
    USART_setup();
    sei();

    while (1)
    {
        /* Replace with your application code */
        USART_retrieve();
    }
}

ISR(ADC_vect){
    USART_transmit(ADCH);
}

```

Code explanation:

Most of the explanation are done in the code snippets via comments, what needs to be made clear is that the program will read the potentiometer from ADC interrupt service routine, and send the serial signal.

3. Problems encountered

Problems encountered during this assignment are as following:

- Equipment problem, the access to second Arduino is hard to reach, the option to buy one is not possible with current financial situation, the practical lessons were non-existent for 2 weeks, and the school was on lockdown, thus the option to work on school is not possible.