Lab 6
UART

Keaton Clark

October 19, 2022

Code

echo2c.c

```c
#include <avr/io.h>
#include <stdio.h>
#include "io.h"

unsigned char U0kbhit() {
    return (unsigned char)uart_available();
}

unsigned char U0getchar() {
    return (unsigned char)uart_read_char();
}

void U0putchar(unsigned char U0pdata) {
    uart_put_char(U0pdata, stdout);
}
```

echo3c.c

```c
#include <avr/io.h>
#include <stdio.h>
#include "io.h"

unsigned char U0kbhit() {
    return (unsigned char)uart_available();
}

unsigned char U0getchar() {
    return (unsigned char)uart_read_char();
}

void U0putchar(unsigned char U0pdata) {
    uart_put_char(U0pdata, stdout);
}
int main () {
    uart_init(9600);
    for(;;) {
        printf("0x%.2X\n", uart_read_char());
    }
}
```

Relavant lines from io.c because I had already written this functionality earlier and made it compatable with the get and put function pointers in C FILE structs which take types "int(FILE*)" and "int(char,FILE*)" respectively.

```c
int uart_put_char(char c, FILE *stream);
int uart_get_char(FILE *stream);
FILE uart_str = {
    .put = uart_put_char,
    .get = uart_get_char,
    .flags = 0b10|0b1, //Read | Write
    .udata = 0,
};
void uart_init(uint16_t baud_rate) {
    stdout = stdin = &uart_str; //set stdout to our FILE stream we made above ^
    UBRR0L = (F_CPU / (16UL * baud_rate)) - 1;
    UCSR0B = _BV(TXEN0) | _BV(RXEN0);
}
int uart_put_char(char c, FILE *stream) {
    if (c == '\n') {
        uart_put_char('\r', stream);
    }
    loop_until_bit_is_set(UCSR0A, UDRE0);
    UDR0 = c;
    return 0;
}
int uart_read_char() {
    loop_until_bit_is_set(UCSR0A, RXC0);
    if (UCSR0A & _BV(FE0)) return _FDEV_EOF;
    if (UCSR0A & _BV(DOR0)) return _FDEV_ERR;
    return UDR0;
}
int uart_available() {
    return UCSR0A & _BV(RXC0);
}
```

Result

From keypress sequence: $abcdABCD < ctrl - c >$