

Austin Bumbalough

CPE 325-08

Lab 08

10/24/2019

Lab 8 Solution

In Lab 8 I configured the UCI peripheral for UART communication and wrote a C program for a chat bot that responds to the wake phrase, "Hey, Bot!".

Output Screenshots

 Output1

Source Code

```
unsigned int delayCount = 0;

void UART_initialize(void);

void UART_sendCharacter(char);

char UART_getCharacter(void);

void UART_sendString(char*);

void UART_getLine(char*, int);

void main(void) {
    WDTCTL = WDT_ADLY_1000; // 1 s interval
    IE1 |= WDTIE; // Enable WDT interrupt
    UART_initialize();
    UCA0CTL1 &= ~UCSWRST; // Start USCI
    _BIS_SR(GIE);
}
```

```

char wakeStr[25];
while(1) {
    UART_sendString("\e[91mMe: \e[0m");
    UART_getLine(wakeStr, 25); // Get line
    while(strcmp("Hey, Bot!", wakeStr)) {
        UART_sendString("\r\n"); // send carriage return and newline
        UART_sendString("\e[91mMe: \e[0m");
        UART_getLine(wakeStr, 25);
    } UART_sendString("\r\n");

    char ageStr[10];
    UART_sendString("\e[34mBot: \e[0mHi! How old are you?\r\n"); // Send c
    UART_sendString("\e[91mMe: \e[0m");
    UART_getLine(ageStr, 10); // Get user's age
    UART_sendString("\r\n\e[34mBot: \e[0m");
    if (!(strcmp("1000", ageStr))) { // If user enters "1000"
        UART_sendString("That cannot be true!\r\n");
    } else {
        UART_sendString("You are so young! I am 1"); // Respond to user ag
        UART_sendString(ageStr);
        UART_sendString(" years old.\r\n");
    }
}
}

void UART_initialize(void) {
    UCA0CTL1 |= UCSWRST; // USCI SW Reset
    P2SEL |= BIT5 + BIT4; // Enable UART pins
    UCA0CTL0 = 0;
    UCA0CTL1 |= UCSSEL_2; // Set clock to SMCLK
    UCA0BR0 = 54; // Sets Baud rate to 19200 bps
    UCA0BR1 = 0; // 2^20/19200 = 54r5
    UCA0MCTL = 0x0A; // UCBSR0 = 5, UCOS16 = 0
}

void UART_sendCharacter(char c) {
    while (!(IFG2 & UCA0TXIFG)); // Wait until Tx buffer is ready to receive c
    UCA0TXBUF = c; // Move c into Tx buffer
    delayCount = 0;
}

char UART_getCharacter(void) {
    while (!(IFG2 & UCA0RXIFG)); // Wait until a character is ready to be reac

```

```

    delayCount = 0;
    return UCA0RXBUF;
}

void UART_sendString(char* string) {
    unsigned int i = 0;
    while (string[i] != (char) NULL) {
        UART_sendCharacter(string[i++]); // Send each character in string
    }
}

void UART_getLine(char* buffer, int limit) {
    char c = UART_getCharacter();
    unsigned int i = 0;

    while ((c != '\r') & (i < limit-1)) {
        buffer[i++] = c; // Store received character in receive buffer
        UART_sendCharacter(c); // Echo character back
        c = UART_getCharacter(); // Get next character
    }
    buffer[i] = (char) NULL; // Terminate string with null character
}

#pragma vector=WDT_VECTOR
__interrupt void WDT_ISR(void) {
    if (delayCount++ > 15) { // 15 second delay
        UART_sendString("\e[34m\r\nBot: \e[0mIs anybody here?\r\n");
        UART_sendString("\e[91mMe: \e[0m");
        delayCount = 0;
    }
}

```