Lab 1 - Pre-Lab & Pseudo Code

Caleb Farnie, Noah Terry

Introduction

This lab will allow us to communicate with the car for future labs. We will be able to send messages to and from the robot car which will be very helpful for debugging future labs. We will also need to communicate with the robot car to control its movement in lab 5. Additionally, we will be able to implement an event driven framework where the messages will be the events.

Pseudocode

Functions in SerialIO.h/c

```
FUNCTION usb_send_byte( byte )
    DESCRIPTION Adds a character to the output buffer
    INPUTS Byte to send
    RETURNS NONE

    CALL push_back() from the homework to add byte to the output buffer

END FUNCTION

FUNCTION usb_send_data( p_data, data_len )
    DESCRIPTION Adds data buffer to the output ring buffer
    INPUTS Pointer to the data to send, length of data
    RETURNS NONE

    FOR index IN data_len
        CALL usb_send_byte() WITH p_data AT index AS INPUT
    END FOR

END FUNCTION

FUNCTION usb_send_str( p_str )
    DESCRIPTION Adds a null-terminated c-string to the output buffer
    INPUTS Pointer to c-string to send
    RETURNS NONE

    SET index TO 0 to increment in the while loop
    WHILE p_str AT index IS NOT null_character
    DO
        CALL usb_send_byte() WITH character at index AS INPUT
        INCREMENT index BY 1
    END WHILE


END FUNCTION
```

```
FUNCTION usb_send_msg( format, char cmd, p_data, data_len )
    DESCRIPTION Sends data from the car to the host machine using USB message format
    INPUTS
        Format: Defines the interpretation of the data bytes (char, int, unsigned int, etc.)
        cmd: Command that the message is responding to.
        p_data: Pointer to the data that is to be sent.
        data_len: Size of the data that is to be sent.
    RETURNS NONE

    CALCULATE the length of the format string
    CALCULATE the length of the message to be sent

    // Send the data
    CALL usb_send_byte() to send the length of the message
    CALL usb_send_str() to send the format of the message
    CALL usb_send_byte() to send cmd
    CALL usb_send_data() to send p_data

END FUNCTION

FUNCTION usb_msg_read_into( p_obj, data_len )
    DESCRIPTION Populates the object with the data in the receive buffer and removes the
        bytes as they are read.
    INPUTS pointer to object that gets populated with data in receive buffer, length of data
    RETURNS FALSE if the receive buffer does not contain enough bytes to fill the
        container and terminates without reading or removing, otherwise TRUE

    FOR data_len
        CALL pop_front() WITH receive_buffer AS INPUT AND SET TO value
        CALL push_back() WITH p_obj AND value AS INPUTS
    END FOR

END FUNCTION
```

Functions in Lab1_Tasks.h/c

```
FUNCTION Multiply_And_Send(value_left, value_right );
    DESCRIPTION Multiplies two values together and sends the result to the host machine.
    INPUTS Two numeric values, the left value and the right value
    RETURNS NONE

    MULTIPLY value_left and value_right together AND STORE AS product
    CALL the usb_send_msg function to send the product calculated in the previous step

END FUNCTION
```

```
FUNCTION Divide_And_Send(float value_left, float value_right )
    Same as Multiply_And_Send(float value_left, float value_right ) but with division.
END FUNCTION

FUNCTION Add_And_Send(float value_left, float value_right );
    Same as Multiply_And_Send(float value_left, float value_right ) but with addition.
END FUNCTION

FUNCTION Subtract_And_Send(float value_left, float value_right );
    Same as Multiply_And_Send(float value_left, float value_right ) but with subtraction.
END FUNCTION
```

Planned Schedule and Task Assignment

We will start by pair programming on the weekend of January 1/27. Here we will begin implementing our pseudo code and making changes as necessary. We plan on spending time Monday, Wednesday, and Thursday continuing our work on lab 1. We plan on finishing the majority of the lab during the week and will then spend the weekend debugging the code. Responsibilities will be shared between the two partners. We plan to pair programming for the entirety of the lab, such as to provide the fairest mode of collaboration.