ME218C Final Project, 2016
Team 6
UART Lib Pseudocode

UART Lib
Module Variables: tx1_data, tx5_data, tx1_flag, tx5_flag

InitUART
Takes nothing, returns nothing

Initialize UART1
Initialize UART5

End InitUART

InitUART1
Takes nothing, returns nothing

Enable the clock to the UART module 1
Enable clock to the GPIO port C
Wait for GPIO port C to be ready
Configure GPIO pins for in/out/drive-level/drivetype
Set GPIO PC4 and PC5 as digital
Set GPIO PC4 as input (receive) and PC5 as output (transfer)
Set mux position on GPIOPCTL to select UART alternate function on C4, C5 (Tiva p.1351)
Mux value = 2 (p.895) offset mask to clear nibbles 4 and 5

Disable UART module 1
Baud Rate = 9600, Integer = 260, Fraction = 27
Write the integer portion of the BRD (260)
Write the fractional portion of the BRD (27)
Write desired serial parameters to the UART line control
Configure UART operation
Locally enable interupts for UART receive interrupt mask (RXIM)
Set NVIC enable for UART1 (see Tiva p.104)
Make sure interrupts are enabled globally
Enable UART module 1

End InitUART1

InitUART5
Takes nothing, returns nothing

Enable the clock to the UART module 5
Enable clock to the GPIO port E
Wait for GPIO port E to be ready
Configure GPIO pins for in/out/drive-level/drivetype
Set GPIO PE4 and PE5 as digital
Set GPIO PE4 as input (receive) and PE5 as output (transfer)
Set mux position on GPIOPCTL to select UART alternate function on E4, E5 (Tiva p.1351)
Mux value = 2 (p.895) offset mask to clear nibbles 4 and 5
Disable UART module 5
Baud Rate = 1000, Integer = 2500, Fraction = 0
Write the integer portion of the BRD (260)

```
Write the fractional portion of the BRD (27)
Write desired serial parameters to the UART line control
Configure UART operation
Locally enable interupts for UART receive interrupt mask (RXIM)
Set NVIC enable for UART5 (see Tiva p.104)
Make sure interrupts are enabled globally
Enable UART module 5

End InitUART5
```

SendUART1

```
Take nothing, returns nothing

Call the ISR for UART1

End SendUART1
```

SendUART5

```
Take a data byte to send, returns nothing

Set tx5 data var
Set tx5 flag
Call the ISR for UART5

End SendUART5
```

UART1ISR

```
Takes nothing, returns nothing

If the interrupt is from RX
      Clear flag
      Post event to Comm SM with the received data as param
If the interrupt is from TX
      Clear flag
      Get next data byte from Comm SM and put in send register

End UART1ISR
```

UART5ISR

```
Takes nothing, returns nothing

If the interrupt is from RX
      Clear flag
      Post event to Comm SM with the received data as param
If the interrupt is from TX
      Clear flag
      If we haven't already sent the tx5 data byte
            Load the tx5 data byte

End UART5ISR
```