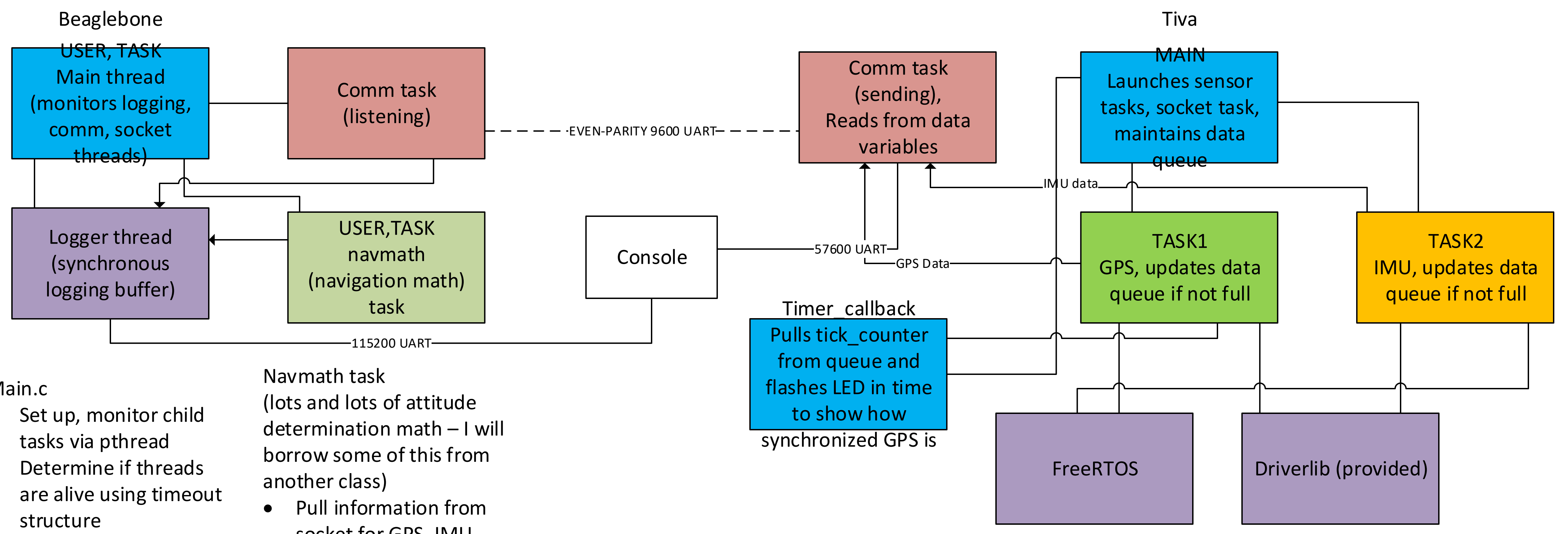
4/8/2018

ECEN 5013
Advanced Practical Embedded Systems

Project 2 Preliminary Software Architecture Diagram

Project: Simple Rangefinding Transponder (Homing Beacon)

Zach Farmer



Beaglebone
USER, TASK Main thread (monitors logging, comm, socket threads)
Comm task (listening)
Logger thread (synchronous logging buffer)
USER,TASK navmath (navigation math) task
Console
115200 UART
EVEN-PARITY 9600 UART
Tiva
Comm task (sending), Reads from data variables
MAIN Launches sensor tasks, socket task, maintains data queue
IMU data
57600 UART
GPS Data
TASK1 GPS, updates data queue if not full
TASK2 IMU, updates data queue if not full
Timer_callback
Pulls tick_counter from queue and flashes LED in time to show how synchronized GPS is
FreeRTOS
Driverlib (provided)

Logging task
- Read in from logging queue, dump to file, dump to terminal in verbose mode
- All threads can write to logging queue via a logging call which formats similar to printf and dumps into the logging queue asynchronously

GPS
- gps_calc

calculation of distance, angle
- gps_dist

calculation of distances to specified targets for testing, field verification
- Test

Calls to unit tests to verify parsing algorithm from GPGGA, IMU data, packet as well as functional tests for distance and angle functions. Includes positive and negative testing. (altitude calculation failure example)

Main.c
- Set up, monitor child tasks via pthread
- Determine if threads are alive using timeout structure
- Set up shared data structures

Comm task
- Polls for, retrieves, and logs packets from Tiva including board information, errors, data, and heartbeat interval. Triggers heartbeat failure error in nav task if a message is not received in correct time

Uart_driver.c
- Provides polling uart interface, spools until finds header symbol from packet and passes to parser

Parser.c
- Parses received packet into GPGGA and imu structs

Print_sync
Print to the terminal in a mutex-guarded, thread-safe manner

Navmath task
(lots and lots of attitude determination math – I will borrow some of this from another class)
- Pull information from socket for GPS, IMU
- Using GPS fix, compare location and direction facing with onboard-stored locations and report whether the rangefinder is pointing at a designated target,
- For example, by triangulating compass direction in degrees with gps fix, we have position and bearing
- By comparing gathered GPS fix data with a known waypoint target, we can determine what direction the waypoint target is in
- By comparing the degree heading of the target and degree heading of our compass with a small margin of error, we can determine if we are facing the target if we are moving quickly enough
- Use accelerometer readings to determine the pitch of the antenna and whether a fix is likely (antenna needs to be adjusted)

GPS task
- Initialize GPS
- Initialization test
- Turn on GPS
- Activate GPS NMEA SAP protocol
- Obtain GPS fix (longitude, latitude, altitude, time)
- Transmit fix data to comm task via shared memory
- Call comm task
- sleep

IMU task
- Turn on IMU
- Assert startup test values (WHO_AM_I)
- Obtain compass, magnetometer, rate gyro, and accelerometer readings inside semaphore (happens very quickly)
- Transmit needed data to comm task via shared memory
- sleep

Utility Driverlib UART driver for GPS
- Polling -driven
- Even parity bit for communication checksum

Utility Driverlib I2C driver for IMU
- Polling-driven

Main task
- Initialize two sensor tasks and comm task
- Initialize semaphore for high-rate IMU task
- Initialize queue for LED synchronization callback
- Start 250ms timer

Timer callback
- Reads from tick_timer queue and toggles LED 2 based on this value
- Shows difference in timing between onboard timer and GPS time

Comm task
- Run on 1.5 second timeout TaskNotify

On notify from GPS
- Take semaphore
- Parse board, software version, hardware ID, and imu and gps data into uart packet structure
- Send to BBB over even-parity UART

On notify from IMU
- Restart IMU task

On timeout/lockout/sensor fault
- Send error message and soft reset tiva