Recitation #11

18-649 Embedded System Engineering

Friday 7-Nov-2014

Electrical & Computer ENGINEERING

Carnegie Mellon

Note: Course slides shamelessly stolen from lecture

Announcements and Administrative Stuff

- **Project 11 has already been posted**
  - Network Scheduling
- **proj11_min_req.xls is also up now**
  - Whoops

Unit test and Integration test summary files

- **Are essential to ensure that our scripts run – need to be present and properly formatted.**
- **These two files are –**
  - integration_tests.txt and,
  - unit_tests.txt
- **These are linked into your portfolio page as well (*Unit Test Summary File* and *Integration Test Summary File*), refer to the sample portfolio for the format -**

  https://www.ece.cmu.edu/~ece649/project/portfolio/portfolio_template/portfolio.html
- **If you have multiple test files for a unit test – ensure that you have one line for each test file**
- **The TAs are not obliged to run your tests manually if these scripts do not work.**

Project 11

- **Complete Network Schedule**
- **Meet Bandwidth constraints**
  - Modify payload translators
- **All unit tests must pass**
- **Create integration tests**
  - Integration tests must run, but do not have to pass
- **Run acceptance tests**
  - Acceptance tests must run (kind of), but do not have to pass.
  - It’s okay for your design to E-Brake, for this project.
- **Update traceability**

Message Dictionary

Message Dictionary					
Sender Node Name	**Message Name**	**Deadline (ms)**	**Message ID**	**Sender Node Type**	**Replica**
Safety Sensor	Emergency Brake	50	1000	20	none
At Floor Sensor	AtFloor				
	Hoistway Limit				
	Hall_Light				
	Hall_Call				
	Drive_Speed				
	Drive_Command				
	Door_Motor_Command				
	Door Reversal				
	Door Opened				
	Door Closed				
	Desired_Floor				
	Desired_Dwell				
	Car_Position				
	Car_Light				
	Car_Lantern				
	Car_Call				
	Car Weight Alarm				
	Car Weight				
	Car Level Position				

Message Dictionary

Message Dictionary					
Message Name	**Deadline (ms)**	**Message ID**	**Sender Node Type**	**Replication Type**	**Base CAN ID**
Emergency Brake	50	1000	20	none	0x0BE81400
AtFloor					
Hoistway Limit					
Hall_Light					
Hall_Call					
Drive_Speed					
Drive_Command					
Door_Motor_Command					
Door Reversal					
Door Opened					
Door Closed					
Desired_Floor					
Desired_Dwell					
Car_Position					
Car_Light					
Car_Lantern					
Car_Call					
Car Weight Alarm					
Car Weight					
Car Level Position					

Network Analysis

Sender Node Name	Message Name	Base CAN ID	Replication Count	Deadline (ms)	Desc
Safety Sensor	Emergency Brake	0x0BE81400	1	50	Ebrake
At Floor Sensor	AtFloor				
Level Sensor	Level				
	Hoistway Limit				
	Hall_Light				
	Hall_Call				
	Drive_Speed				
	Drive_Command				
	Door_Motor_Command				
	Door Reversal				
	Door Opened				
	Door Closed				
	Desired_Floor				
	Desired_Dwell				
	Car_Position				
	Car_Light				
	Car_Lantern				
	Car_Call				
	Car Weight Alarm				
	Car Weight				
	Car Level Position				

Network Analysis Continued

	Field 3			Total Payload Bit Len	Total Payload Byte Len	Best Case Msg Len	Worst Case Msg Len	Best Case BW (bits/sec)	Worst Case BW (bits/sec)
Bit Len	Desc	Type	Bit Len						
0	n/a	n/a	0	1	1				
							Total bw		

Reducing bandwidth consumption

- **Up till now, simulator has run with unlimited bandwidth**
  - Previously it was OK to send a boolean value in a 64-bit payload
- **Now bandwidth is limited to 200,000 bits per second (this value will not be changed in the future)**
  - Need to modify data payloads to reduce bandwidth consumption
- **Use "-b 200" for acceptance tests**
  - You do not need to use this for integration and unit tests
- **Computing best case and worst case bandwidth**
  - Best case is no stuff bits
  - Worst case is one stuff bit per every four data bits
  - For a refresher check out the formula in Lecture 12 - CAN Performance
    - Use this formula for the most conservative upper bound

Ways to Reduce Bandwidth Consumption (1/2)

- **Ensure that you have a single message dictionary**
- **Remove unused message types**
  - E.g., mHallLight (remember to remove it from all other documentation too! For example - message dictionary in Requirements I, updating removed messages and the input and output interfaces in the Requirements II)
- **Combine message types**
  - Must have same period
  - Must originate from same instance of a node
    - You can combine multiple messages originating from the same dispatcher
    - But you can't combine messages from four different door controllers
- **Use the minimum number of bits to send values**
  - CAN specification requires payloads to be in bytes
  - Takes 1 byte to send an 'on' or 'off'
    - 7 unused bits, so you could send other messages in these
  - How many bits do you need to send the drive speed? Hint: Its not 64

    (Keep an eye the number of bits being used for it)

Ways to Reduce Bandwidth Consumption (2/2)

- **Issue log update**
  - If you end up removing a message, ensure that the issue log is updated.
- **If no module actually reads a given message, its generally safe to remove it.**
- **You may NOT add new messages to the schedule unless you obtain approval from the course staff.**
- **You may NOT change message periods unless you get prior approval from one of the TA's (Preferably in an email copying the group & staff mail list)**
- **You may NOT use bits of the message ID to transmit data values (limitation of the simulator architecture)**
- **You may NOT remove any of the pre-defined constants. Doing so may cause your code to no longer be compile compatible with the simulation framework.**

Questions?