EE 491 Senior Design Group Meeting

Safe Communication Between Lead and Following Vehicle

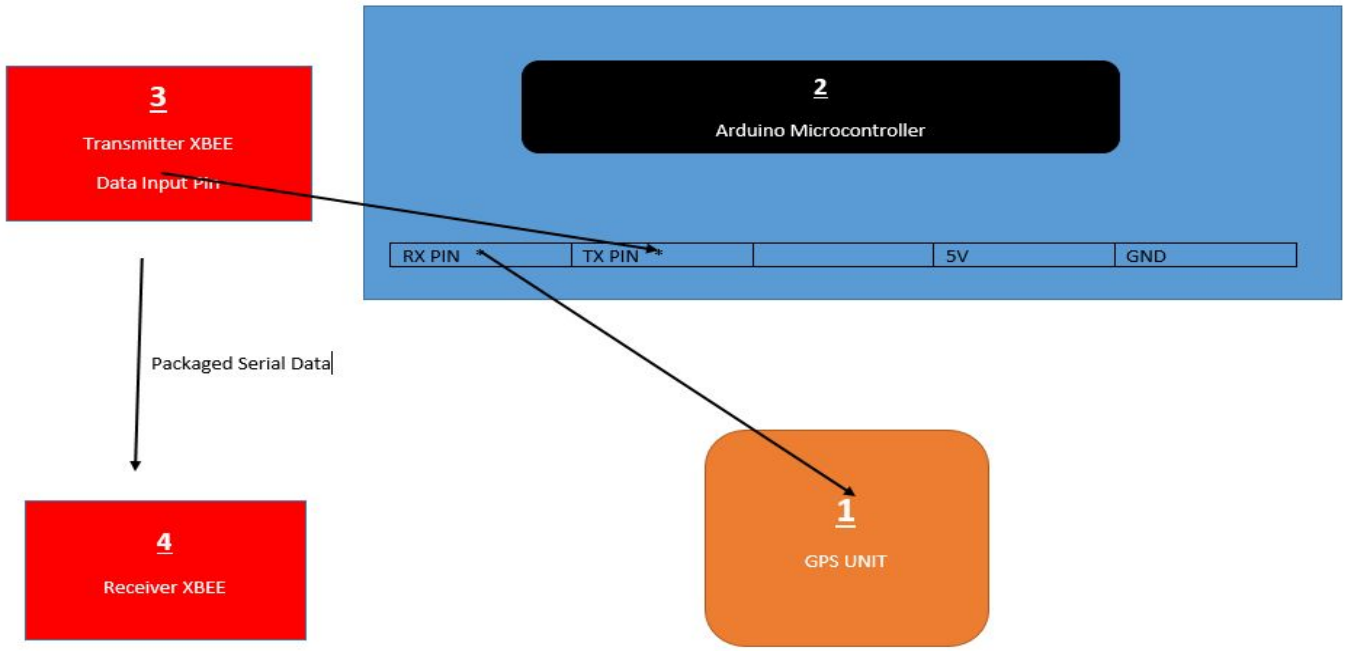
Week 10 Report

Team Members:

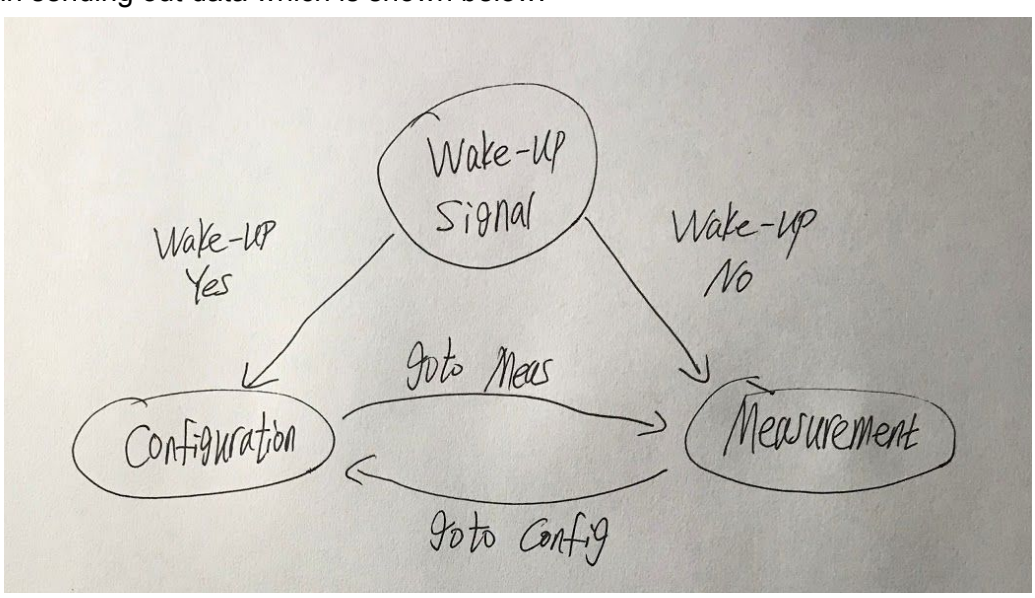
Bradley Stiff- Software Lead, Project Lead
Justin Wheeler- Software Lead
Sanguk Park- Scribe Lead, Communication Lead
Zhize Ma- Scribe Lead, Hardware Lead
Junho Chun- Hardware Lead
Yifan Lu- Hardware Lead
Jose Candelario- Project Lead, Communication Lead

This Week Accomplishments:

This week, we made regular visits to Intrans to start gathering data from the GPS. We implemented our arduino code and connected the transmitter Xbee to the TX pin of the arduino UNO. The RX pin of the arduino was connected to the data pin of the gps. How this circuit operated was the GPS would send the serial data to the arduino which the RX pin (Receiving Pin) would receive that serial data and then convert that data into 16-bit packages through the microcontroller of the arduino and that converter would then go through the TX pin (Transmitter Pin) to the transmitter Xbee. A more detailed schematic is shown below which shows the function of the circuitry of our overall transmitter. From the schematic, the numbers indicate the order in which the data goes through. Using this transmission circuit, we tested the efficiency of data transmission at the GPS units used on the following car.



In order to make the XBEE compatible with the GPS, we had to observe several things on the communication protocols of the GPS. In our previous attempts to gather data, we ran into certain technical limitations and therefore, had to make certain adjustments to make the hardware compatible with the GPS. We knew since we had to make hardware adjustments like the schematic shown above, we would have to pay attention to the data protocols in which the GPS output the data. In realized that the GPS's communication protocol had certain procedures in sending out data which is shown below:



The serial data format that would be transmitted by the GPS also had its own message structure in which we would have to adapt our microcontroller to. The raw data that we first got from our GPS had a format which was unexpected.

PREAMBLE	BID	MID	LEN	DATA	CHECKSUM

The above is how we expected data to be received from the GPS. Once we received the serial data the first four characters should 0xFAFF as part of the Preamble and Bid. Then Mid gave us the message identifier followed by the len that would give us the length of the message. The Data would be given in relation to length. Which in this case would be the raw serial code. Checksum would then add the total bytes in the previous five sections and give us the sum.

The GPS serial data had an individual format in which organized that data in fields with each field having an independent width. The first raw data that we collected was different from what we expected in which we only depended on the latitude and longitude of the GPS, but that was not the case. We later realized that the data that appeared on the receiving end of the XBEE was coded in ascii which we would have to decode to hex for each of the characters. Due to the data format of the serial data of the GPS, we had to make major changes to our arduino code so that the data sent from the GPS can be encoded and then sent to the receiving XBEE.

Individual Contributions (10/21~10/28)

Team Member	**Contribution**	**Weekly Hours**	**Total Hours**
Brad Stiff	Helped optimize code to read characters transmitted from the GPS. Researched the different GPS transmission modes, to ensure output transmission.	5	40

Jose Candelario	Researched the form in which data was transferred from the GPS. Created some Ideas on how we could make sure that we were getting what we expected from the GPS.	3	50
Junho Chun	Considered how to power for each sensor, and all components from powering device.	3	33
Justin Wheeler	Wrote Python scripts to both receive and send the GPS data. Replaced the arduino with a Raspberry Pi to read serial data directly from the GPS USB cable. Started looking into hooking the Xbee into the NVIDIA Drive PX 2.	6	35
Sang Uk Park	Tested out several methods to receive the GPS data using an arduino. Most failed except when we used our own computers. We plan on switching from arduino to raspberry pi.	4	41
Yifan Lu	Verified the connector's pins assignments of GPS and tried to receive on the python script.	3	34
Zhize Ma	Help on hardware part of arduino, make sure hardware part is stable.	4	35

Goals for Next Week

Make sure that our arduino code can efficiently and successfully encode the serial data sent to from the GPS and have all of the outgoing data be consistent as possible. Overall, we will need to go to INTRANS more often so that we can have more immediate feedback with our progress and find any problems that may intercept the data that we collect.