#**FullCommunication.ino**

The full communication protocol for Team Sai Cheemalapati.

The Arduino code can be found as
src/BaseCommunication/FullCommunication/FullCommunication.ino
The code contains everything needed to finish the last part of the IDC.

#**What happens**

To communicate, each bot holds a list (array) of which order numbers have been declared by the other bots and which order numbers have left their mark and are moving towards the end. When a bot first finds its number, it begins to send information as little packets to all the bots. The packets take the form of of the start character "=", a single character data payload, and a checksum character. The data payload is a letter from g-k, and the checksum is a simple hash of the first to characters, in hexadecimal form. Thus, the program filters out any letters that aren't '=', a-k, or A-K. Any packet with an invalid hash is also thrown out. When a bot receives a valid packet, it updates it's "database" using the received data. When the bot reaches the black middle line, the bots then use that data to decide when to go. The bots are made to go under the following conditions, in this order:

1. Your bot is the first bot. It goes automatically
2. The bot in front of yours has gone.
3. The bot in front of yours hasn't gone, but the bot in front of them has, and it's been over 30 seconds. We assume that either they went mute or the bot died, so we go anyways.
4. Your bot is bot #2, and it has been over 30 seconds since #1 left

Lastly, if for some reason no condition to leave presents itself, the bots have a fallback timer that will let them leave after a certain amount of time, depending on

their order.

The protocol also allows for one bot to either fail to obtain its number or for two bots to falsely choose the same number. If there is such a collision, the bot who found its number first will keep it, and the second will throw its away. Then, if every bot but one has a number, the remaining bot will deduce its position.

How to Copy

There are three parts of the code you will need to copy to your own sketch.

1. Everything before setup(), copy it to before your own setup()
2. The Xbee.begin() within setup()
3. Everything after loop(), just put it way down at the bottom of your script, so you don't have to look at it

How to Implement

1. Replace all delay() functions with sDelay(). This allows your bot to communicate even when it's stopped. Just don't do this with the small <5 ms delays when reading the IR sensors, for instance.
2. When your bot knows it's number, call foundOrder([number here]).
3. When you reach the big black line in the middle, run
   int OrderNum = doIGo();
   This will pause your bot until it's time to go and will return your (maybe new) order number.
4. After you are going, call sendMoving(); when you start the curve. This lets other bots know that it's their turn
5. Forgot to add, replace the In/Out pin constants at the very top with your own

Sample Code

```
void loop() {
  sDelay(10000); // Wait at the beginning
  int orderNum = moveAlongLineAndDoMyTaskAndShit();
  foundOrder(orderNum); // Report our number
  doSomethingThatTakesMeToTheBlackLine(); // standard C method for years
  orderNum = doIGo(); // This will pause your bot until it's time to go
  lineFollowACoupleFeet();
  sendMoving(); // Let everyone else know it's their turn
  continueToHash(orderNum); // And you're done.
  while(1) {
    sDelay(1000);
  }
}
```