Intro

For this project we set out with two basic aims. The first was for us to write a program as that the Kinect would detect the movement of a user pass a message to the Arduino. The second was for us to program the Arduino so that it would interpret the commands sent from the Kinect and control the Zumo. We had an aspirational aim of creating a visual on a pc or app displaying a map of where the Zumo had travelled. How or if we have met these aims will be described below. We utilise the following components: 2 laptops, Zumo shield, Microsoft Kinect, Arduino Uno, X-Bee Shield, 3 X-Bee radio modules.

We achieved the first basic aim by sourcing an example of a C# program which showed us how to get the Kinect to detect a body and determine locations of the joints of that body. We then used this to detect the movements of these joints. We adapted this source so that when the joints are in a certain position it sends a command to the Arduino for example when the right hand Y position is greater than the right hip Y position it sends a command to the Arduino of turn right. We originally intended to transfer data over Bluetooth however we decided to change this as we discovered it would be possible to have this functionality using X-Bee shields which we already had knowledge of. It sends a message using a serial port. However, to prevent the serial port becoming over loaded we only send the message if it is not the same as the last. The speed and the command are sent in the same message an example message would be a,50! The speed is calculated by the distance between the two joints.

We achieved the second basic aim which was for the Arduino to accept and interpret the command sent from the Kinect to control the Zumo. To do this the Arduino reads in the command sent over the serial in two parts. The first part it reads is the command letter this will either be w, a, s, d or x and the second part will be the speed at which it should carry out the command. First of all, we use the command letter part of the message to set the direction of the Zumo motors and then we use the speed to set the speed of the motors. However if the speed in the message is greater than 125 we set the speed to 125 to prevent damaging the motors.

We had an aspirational aim of mapping on a screen where the Zumo had travelled we were unable to get this going to due to the inaccuracy and complex compass that the Zumo has. Instead we have messages sent via MQTT to a smartphone, so the user of the phone knows what movements the Zumo is currently making. An example of a message would be ‘Zumo moving forward’. In order to implement this, we have set all the X-Bee’s so that when the X-Bee which is sending messages processed from the Kinect are sent to both the Zumo X-Bee and another X-Bee so that we can use these messages to transform it so that we can send it over MQTT.

* sources for any code used
* an explanation of how that code works
* discussion/description of adaptations made to use the code in your application.