

Team 18 Task Diagram

User Rover

The User Line Sensor is the user rover's line following sensor. It sends to the U Line Sensor Interrupt Handler whether each of its sensors is detecting the line. The U Line Sensor Interrupt Handler is timer driven, and sends the sensor data into the Msg Queue A. Msg Queue A also receives instructions from the User Server Input Interrupt Handler, which is also interrupt driven. The instructions are read from the User WiFly, which is receiving turning directions from the Raspberry Pi. The instructions and sensor data in the Msg Queue will have a tag to identify which type of information the Msg Queue A is passing on to the User Rover Thread. The User Rover Thread will take the directions from Msg Queue A and upon the next intersection the User Line Sensor's data detects, the User Rover Thread will send turning instructions to the Msg Queue B. If the sensor hasn't detected an intersection, the User Rover Thread will pass instructions to continue following the line, with whatever adjustments are needed. Msg Queue B's information is instructions on what direction each tread on the rover should be spinning. That information is sent to the User Rover Interrupt Handler, which is timer driven. Then the User Rover Interrupt Handler tells the User Rover's Physical Device, what the treads should do.

Ghost Rover

The Ghost Line Sensor is the ghost rover's line following sensor. It sends to the U Line Sensor Interrupt Handler whether each of its sensors is detecting the line. The U Line Sensor Interrupt Handler is timer driven, and sends the sensor data into the Msg Queue C. Msg Queue C also receives instructions from the Ghost Server Input Interrupt Handler, which is also interrupt driven. The instructions are read from the Ghost WiFly, which is receiving turning directions from the Raspberry Pi. The instructions and sensor data in the Msg Queue will have a tag to identify which type of information the Msg Queue C is passing on to the Ghost Rover Thread. The Ghost Rover Thread will take the directions from Msg Queue C and upon the next intersection the Ghost Line Sensor's data detects, the Ghost Rover Thread will send turning instructions to the Msg Queue D. If the sensor hasn't detected an intersection, the Ghost Rover Thread will pass instructions to continue following the line, with whatever adjustments are needed. Msg Queue D's information is instructions on what direction each tread on the rover should be spinning. That information is sent to the Ghost Rover Interrupt Handler, which is timer driven. Then the Ghost Rover Interrupt Handler tells the Ghost Rover's Physical Device, what the treads should do.

The Pixy Cam

The PixyCam is a small camera that will look for the two rovers on the field which have specific markings on them. It sends where it sees the markers to the PixyCam Interrupt Handler, which is timer driven. The Msg Queue G reads from the interrupt handler and sends the information to the Pixy Cam Thread to interpret the rover's location and direction in the maze. Then the

coordinates and orientation are passed into the Msg Queue H. PixyCam Interrupt Handler is timer driven and reads the information from the Msg Queue H to the PixyCam WiFly.

Server (Raspberry Pi)

The Wireless Communication is the built in component of the Raspberry Pi that allows it to connect to a network. The Wireless Communication will send directions to the User WiFly and Ghost WiFly, as well as receive the location and direction of the rovers from the PixyCam WiFly. This will be read by the Raspberry Pi's operating system. A Server Input Handler Thread will be used to check on any input from the PixyCam, and send it to Msg Queue E. The Server Main Thread will take the rover's location and orientation from the Msg Queue E and send appropriate directions for the rovers to Msg Queue F. The Server Main Thread reads the rover directions from the Msg Queue F and directs the Raspberry Pi's Operating System to send the appropriate instructions through the Wireless Communications to the respective rover.