

Floor Cleaning Robot

Developers

Steve Antony Xavier Kennedy

Sanju Prakash Kannioth

Github link: <https://github.com/steveantony5/Floor-Cleaning-Robot>

University of Colorado Boulder



Introduction:

Problem:

Most of us in this busy world doesn't have time to clean our house. Also it takes hours to clean our house every week.

Solution:

This project helps to keep your house clean. The floor cleaning robots automatically does the work of cleaning your house floor daily. This helps your precious time in other important tasks

Hardware used:

- BeagleBone Black
- Tiva C series TM4C1294XL – freertos
- Lux sensor – APDS-9301 (<https://www.sparkfun.com/products/14350>)
- Ultrasonic sensor – HC-SR04 (<https://www.sparkfun.com/products/13959>)
- Water level sensor (<https://www.makerlab-electronics.com/product/rain-water-level-sensor/>)
- Relays SPDT (<https://www.sparkfun.com/products/100>)
- Motor driver - L293DNE (<https://www.sparkfun.com/products/retired/14297>)
- Two 5V DC motors
- Robot Chassis
- 12 V Solenoid Valve (<https://www.sparkfun.com/products/10456>)

Modules:

Ultrasonic sensor:

This helps the robot to traverse on the floor. It tries to detect obstacles on its way and changes path based on the obstacles.

Lux sensor:

This helps to detect the time when everyone in the house has slept. If there is no light in the room, the robot automatically switches on and cleans your house floor

Water level sensor:

The robot has a small water tank for cleaning the floor using water. The valve closes if the water level is below the lower limit.

Actuators:

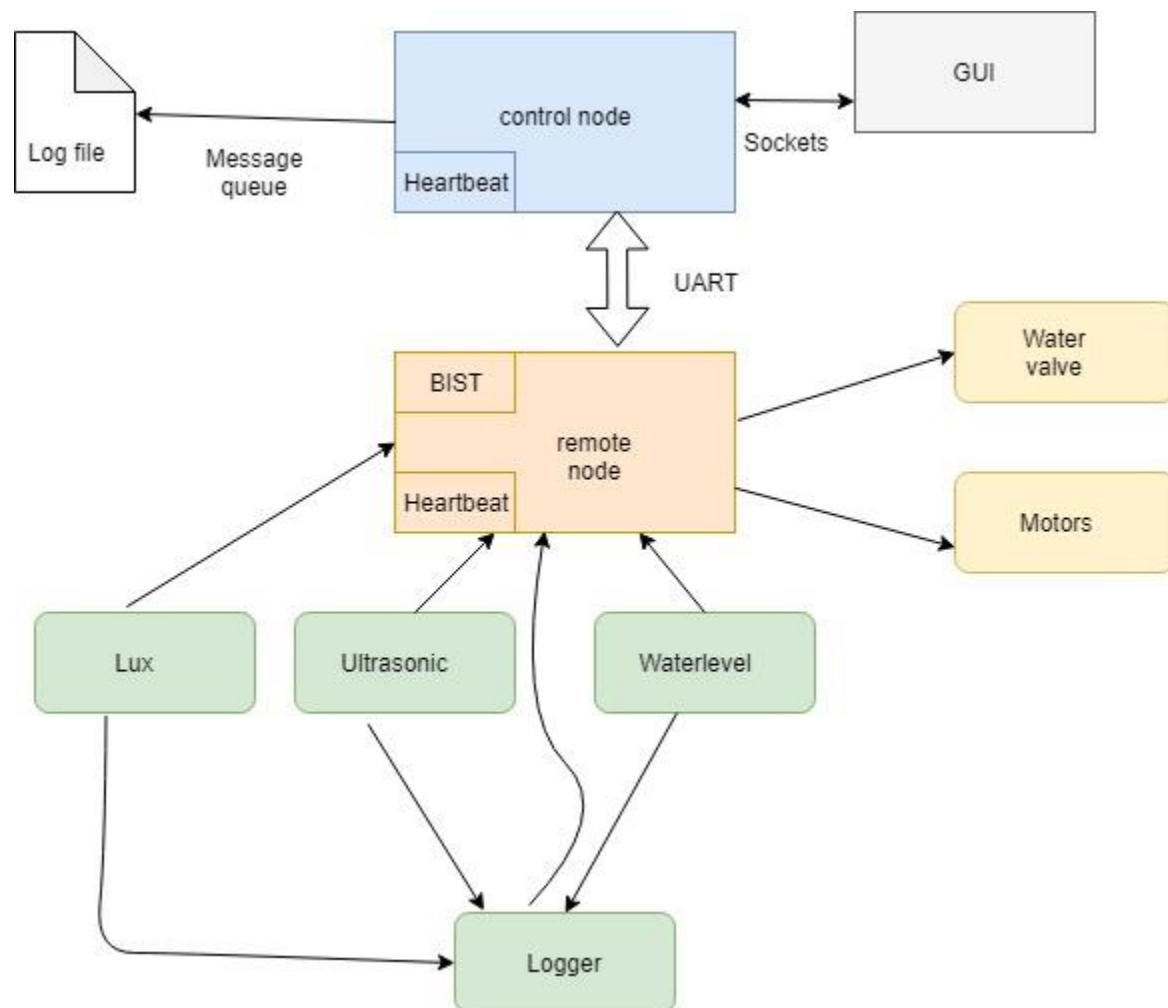
Wheels:

The robot has two wheels which runs based on the input from the ultrasonic sensor

Water valve:

The solenoid valve is activated based on the water level in the tank

Block Diagram:



Product Description and Capabilities:

There are two application modes of operation:

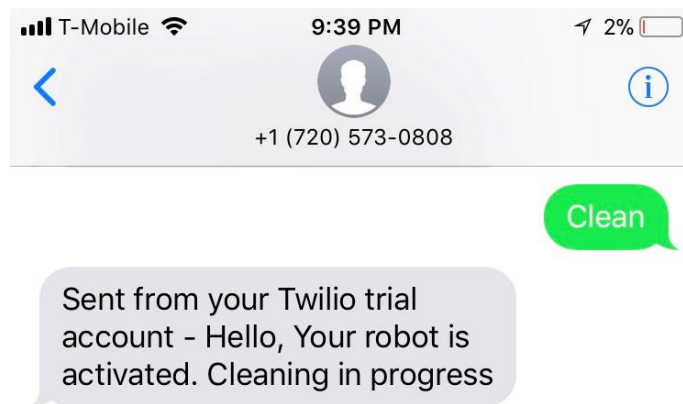
The modes can be chosen on the GUI

1) Auto mode

In auto mode, the robot runs on its own by detecting obstacles using the ultrasonic sensor. There is no human interaction.

There are two ways to start the robot in this mode

- Auto start when the light goes dim
- Send a text message as "Clean" to +1(720)5730808 from any twilio registered number



2) Manual mode

In manual mode, the ultrasonic sensor is disabled. The robot can be controlled using the forward, backward, left, right buttons on the GUI.

Manual mode and auto mode can be switched on back and forth using the GUI.

Stop button on the GUI can be pressed in any mode to stop the robot moving.

Dialog

Floor Cleaner - Makes life easier!

Display

Water level

Lightness level

Obstacle distance

Operational status

Remote Node status

Ultrasonic sensor status

Lux sensor status

Water level sensor status

Valve position

Application Mode chosen

Manual

Auto

Stop

Hand control on manual mode

Forward

left

right

Backward

Levels of operation:

Normal operation:

- When all the sensors are properly connected and are working, the operational mode is “normal” which will be displayed on the GUI in operational status display

Degraded operation:

- When the ultrasonic sensor is disconnected or not working and the control node is working properly, the robot is in degraded mode.
- In this operation, the application mode is automatically switched to manual mode. The robot cannot enter auto mode.

Fail safe operation:

- When the ultrasonic sensor is disconnected or not working and the control node is inactive, the robot goes to fail safe operation and the motors stop irrespective of any application mode it was in previously.

Software Architecture:

Remote node:

The Tiva board is used as the remote node where the sensors are connected.

The sensors connected to the remote node are Ultrasonic sensor, Water level sensor and Lux sensor.

The sensor sends the data to the control node with the help of UART.

The remote nodes have actuators like motors, water valve. These actuators are controlled based on the control signals received from the control node. Under normal operation, the processing overhead is done by the control node. When the control node is inactive, the processing of the ultrasonic data alone is done in the remote node.

Logger:

All the events and abnormal conditions are logged to the logger file on the control node with the help of UART communication.

Control node:

The Beaglebone Black is used as the control node where the processing of the sensor data is performed, and it sends control signals to the remote node. Also, logger is available on the control node too.

Startup tests:

As soon as the remote node is powered on, it performs power on-self test (POST) for the sensors and checks if they are working in good conditions. If not, it logs appropriate error message to the logger file on the remote node.

Closed Loop Control:

Obstacle detection:

When any obstacle is detected using ultrasonic sensor, the control node gives signal to run the robot backward and turn right and move forward.

Water level detection:

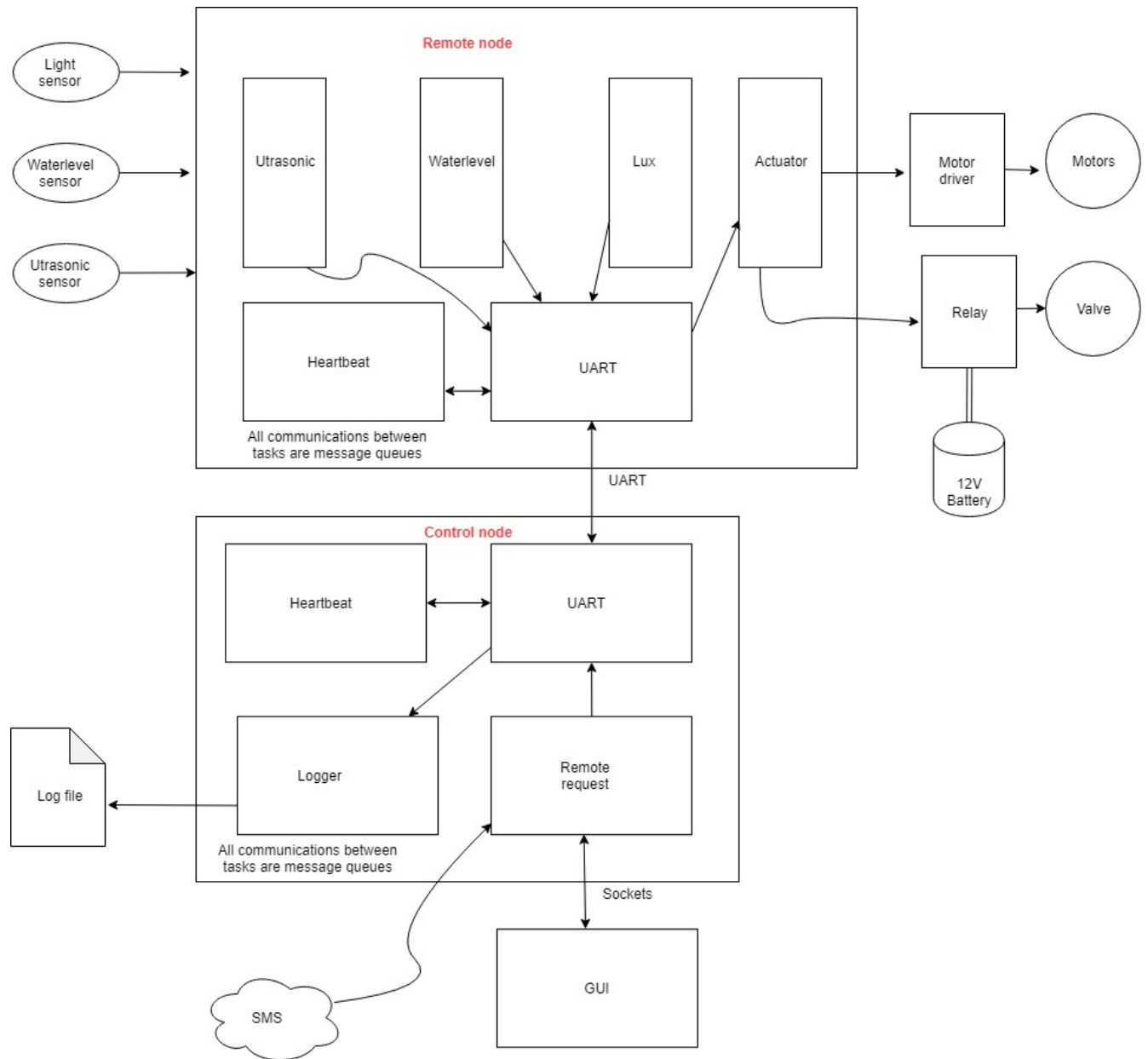
When there is enough water level in the tank, the relay opens the water pipe valve and allows the cleaning to be performed with water.

When there is no enough water level in the tank, the relay closes and doesn't allow water to pass through the pipe

Lightness level detection:

The robot starts automatically in auto mode when the room is dark (i.e during night time). This helps the robot to start each day during night and clean the room

Software Architecture Diagram:



Extra Credits Attempted:

- Control node operational status using GUI
- Control node collecting remote node log after restoration of communication
- Tiva C non i2c Sensor interface

- Tiva C additional sensors beyond two
- Tiva C non-led output devices/ additional output devices: Motors, motor driver, Relay, solenoid valve, wheels

Key Learning Points:

Steve:

- This helped me to learn the real understanding of locks. I learned what would be the issues if there were no locks
- Learned how to create a graphic user interface
- Learned how to interact between python code and C code and the issue that would arise
- Learnt how to interact between or active a task in the application based on a sms

Sanju:

- Integration testing should be done as and when new features are added as it becomes really difficult to debug faults later on.
- Learned to use interact between different programming environments using sockets.
- Learned how to properly use locks in design.
- Learned how to communicate between devices using the UART.

Verification and Validation plan:

https://docs.google.com/spreadsheets/d/1fT7aAccYDtFnekfaoFxfsdTUZt_fwKRsD7TQs9PxbyQ/edit#gid=1734689929