Robot Car Project Report

Nov 2017

* Hardware Specification

1. Ardunino UNO micro controller
2. Geared 5V DC drive Motors x 2
3. Motor Drive Controller, L298N Dual Bridge
4. Microphone Sound Detection Sensor Module
5. Ultrasound Sensor HC-SR04 x 3
6. 8 bit port extender I2C 8974
7. RGB LED x 2
8. LCD display 1602 I2C
9. Motor encoder HR-20 x 2

* Circuit Diagram
* Function Description
  1. Power supply voltage is checked and display by LEDs or reading showing in LCD display. The Robot car is then switch to Standby Mode showing by flashing double blue LED. Standby Mode will last 3 mins and switch to Explorer Mode with flashing double green LED . If any valid clap count detected in Standby mode, Robot Car will switch to Manual Command Mode with flashing double white LED 5 times & all LED off . A valid clap count is less than 4 counts within 250ms of clapping.
  2. In Explorer Mode, Robot Car will behave autonomously. It engages the ultrasound sensor in three direction, front, left and right to achieve avoiding obstacles when travelling. It also utilizes the motor encoder to keep straight forward travel.
  3. In Manual Command Mode, Robot Car will only move from manual commanding by responding to clap count detected as listed as:
     1. One count for motor turn forward
     2. Two count for motor reverse
     3. Three count for motor move in southern cross figure
     4. Four count for motor move Zig-Zag
  4. In Explorer Mode, any clap detected will return to Standby Mode. In command Mode, without any clap detected will return to Standby Mode
* Installation and Coding
  1. Motor drive Board
     1. Initially engage the Enable PIN A and B for speed control of Motors , but finally with bypassing those pin to save GPIO access of the Arduino UNO board.
     2. Instead of direct speed control from PIN A and B, IN1 & IN3 for Motor turning direction and IN2 & IN4 for speed.
     3. Speed Control is by PWM output from UNO, but found the “reverse” direction speed will be (255-PWM) to achieve the forward PWM value.
     4. Disadvantage of power consuming as motor control ON at all time is noticed.
  2. Motor Encoder
     1. As the efficiency and accuracy of drive motors is not enough to keep the car driving in straight line, motor encoder is engaged to detecting the rotation of motor with slotted wheel mounted on each motor.
     2. With controlling Left and right motor PWM value with the reference of encoder reading, the motor drive left and right can be balance to achieve straight line movement.
  3. Microphone sound sensor
     1. It is used to communicate to the Robot Car control. Digital output is connect to the Interrupt PIN (#2) of UNO. The board listens to any sound, hand clapping and trigger the interrupt.
     2. The clap count will be counted and directed to different mode for the Robot Car activity.
     3. Activity mode are manual command mode and self explorer mode.
  4. Ultrasound sensor
     1. Three ultrasound sensor are set up at the front, left and right direction in the Robot car
     2. It is only used in the self explorer mode to avoid obstacles and prevent crushing.
     3. Their Trigger pin is connected to separate PIN (#8, #10, #11) in UNO, but the Echo pin are connected together with diodes and pull up resistor to interrupt PIN #3 of UNO.
     4. The ultrasound sensors return the distance at all time.
  5. RGB LED
     1. Two RGB LED are used to display the status of the Robot Car as the response of the Clap communication.
     2. Any direct clap communication will need a confirmation clap command to execute the command. For example, in manual command mode, a motor move forward command is received by one clap, the LED in the BLUE color will be flushing for 5 sec waiting for another confirmation clap before forward command is executed.
  6. I2C port extender
     1. Due to each RGB LED require three GPIO, the extender can have extra 8 ports for it. Also it is I2C communication and only use the GPIO A4 and A5.
  7. I2C LCD display
     1. It connected to I2C circuit to save GPIO on UNO.
     2. It is programmed to display status of Robot Car.

Conclusion

During the project we experience different hardware configuration to achieve the designed function as well as improving the design of function. We tried out the two approach of programming to adapt the hardware requirement. The traditional Main Loop approach limited our achievement of real time data programming such as the clap detecting by microphone, the ultrasound sensor and motor encoder. But task orientated real time FreeRTOS programming exceeded the capability of the Arduino UNO in memory and processing power. At the end, we have to combine both approach.

The poor performance the 5V drive motor and the plastic steering wheel costed us a lot of program coding trouble to achieve straight line travel. But finally it is resolved with a wheel encoder and single ball bearing front wheel.

We found it is important to co-operate the hardware and software (the programming) from the design.