GROUP 9 -Anonymous

Hardware Game: Car Game Controller

Devices used:

1. Arduino -1
2. Raspberry Pi -1

Sensors used:

1. Potentiometer
   Quantity: 1
   Description: The value of the potentiometer is used for steering wheel
2. IR Proximity Sensor
   Quantity: 1
   Description: The value of the IR proximity sensor is used for brake
3. Ultrasonic Distance sensor
   Quantity: 1
   Description: The value of the Ultrasonic Distance sensor is used for accelerator
4. Thumb Joystick
   Quantity:2
   Description: One joystick is used for gearbox. 6 different buttons are enable for the gears based on the angles formed with X and Y axis. Another joystick is used for additional features that can be configured according to the game.
5. Buzzer
   Quantity:1
   Description: When the score becomes 0 or the game ends the buzzer is triggered for 1 second and score is published through mqtt.
6. Push Buttons
   Quantity:2
   Description: One is used to indicate the end of the game and other is used for resetting the score.

SETUP

For the Car Controller

1.For uploading the hex file install flip and Atmega driver

2.Arduino code should be loaded into Arduino using serial hex file.

3.Now,the Arduino game controller hex file should be loaded into Arduino in the DFU mode.

4.Now,we can start a game and in the game controller settings change the settings for each control based on the sensors value.

5.For particular games ,in x360ce application mapping should be done for Unojoy buttons to corresponding Xbox controls.

6.Start playing your game.

7. We have 2 hex files one for serial mode and another for game controller

8. We have 2 code files

a. Unojoy.h which contains the definition of the struct dataForController_t and contains function like getBlankDataForController(used for initialising all the elements inside the struct) and setupUnoJoy(to set up the serial communication)

b.UnojoyArduinoSample.ino contains setup function to setup the Arduino ports Unojoy and getControllerData function to set the values in the struct dataForController_t according to the sensor values and a loop function.We hardcode the rightstickX to 512 as we are not using it for anything and it is the default value.

For the MQTT part and Game Part

1. Install mosquitto and paho mqtt in the raspberry pi.
2. The timer is started by pressing the reset button .
3. When the actual computer game ends we end the game by pressing the end button
4. We have 2 python codes.
   a. 1 python code is the code for the game where publishing also takes place using mqtt. We have hardcoded the initial score and whenever brake is applied the score is decreased proportional to the time for which it is pressed. If the game completes we press the end button to get our final score.
   b. 2nd python code is for subscribing to the mqtt topic through terminal of another computer.

For MQTT in android phone

Install MQTT dashboard in your mobile and subcribe to the topic to get the scores