NodeMcu ROS Project

These instructions and information are written by me(hmet cagri aksoy). The project is about ROS Controlled NodeMCU Sensor read and servo control. This project is prepared for robotics final lecture which is ongoing master deggre with EE7055 ID on Marmara University.

Preliminary Information

Nodemcu

NodeMCU is a low-cost open-source IoT platform. It initially included firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems and hardware which was based on the ESP-12 module.Ref

ROS

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. Why? Because creating a truly robust, general-purpose robot software is hard.Ref

Rosserial

ROS Serial is a point-to-point version of ROS communications over serial, primarily for integrating low-cost microcontrollers (Arduino) into ROS. Ref

MQ sensors

Detecting and measuring pollution and the main gases is always relevant for a while it is possible to find low-cost, reliable sensors for sale: that's the ones from the MQ series. MQ sensors can detect these type of gases: * MQ-2 = flammable gases such as LPG and propane;

- MQ-3 = ethanol;
- MQ-4 = methane (CH4) and natural gas;
- MQ-5 = LPG and methane;
- MQ-6 = LPG and methane;
- MQ-7 = carbonic monoxide (CO) and hydrogen (H2);
- MQ-8 = hydrogen (H2);
- MQ-135 = gaseous ammonia (NH3), benzene, ethyl alcohol and carbonic dioxide (CO2).REF

Environment

I have used the Arduino IDE to embed the rosserial to Arduino based nodemcu. I have worked on ubuntu 18.04 LTS. My ros version is melodic morenia. Board, sensors, and servo:

ESP board's information and configuration are not valid on Arduino by default installation. First of all, I have installed that, and Rosserial is included on Arduino IDE from adding the library section. Then, the sensor value read and servo control codes are written. In Ros, nodes and topics are important. We are sending the sensor data under topics via signals. To control the servo, we are waiting for the signal from the master which represent our Linux machine. # The file of mq_sensor_ROS includes MQ-* sensor configuration with nodemcu board and rosserial features. The file of mq_servo_ROS includes MQ-* servo motor configuration with nodemcu board and rosserial features.

```
 // Sensor with Buzzer example by Cagri.Aksoy
#include <ESP8266WiFi.h>
#include <ros.h>
#include <std_msgs/String.h>

ros::NodeHandle  nh;

std_msgs::String str_msg;
ros::Publisher chatter("chatter", &str_msg);

char hello[13] = "hello world!";
// threshold value
int sensorThres = 400;
int Led = D1;
int smokeA0 = A0;
int buzzer = D2;
void setup()
{
  Serial.begin(57600);
  nh.initNode();
  nh.advertise(chatter);
}

void loop()
{
  int analogSensor = analogRead(smokeA0);

  Serial.print("Pin A0: ");
  Serial.println(analogSensor);
  // Checks if it has reached the threshold value
  if (analogSensor > sensorThres)
  {
    digitalWrite(Led, HIGH);
    tone(buzzer, 1000, 200);
  }
  else
  {
    digitalWrite(Led, LOW);
    noTone(buzzer);
  }
  delay(100);
  str_msg.data = hello;
  chatter.publish( &str_msg );
  nh.spinOnce();
  delay(1000);
}
```

```
 servo control example
#define ESP8266
#include <Servo.h>
#include <ros.h>
#include <ESP8266WiFi.h>
#include <std_msgs/UInt16.h>

ros::NodeHandle  nh;
int Servo_pin D1;
Servo servo;

void servo_cb( const std_msgs::UInt16& cmd_msg){
  servo.write(cmd_msg.data); //set servo angle, should be from 0-180
  digitalWrite(13, HIGH-digitalRead(13));  //toggle led
}
ros::Subscriber<std_msgs::UInt16> sub("servo", servo_cb);

  void setup(){
  pinMode(13, OUTPUT);

  nh.initNode();
  nh.subscribe(sub);

  servo.attach(Servo_pin); //attach it to pin Servo_pin
}

void loop(){
  nh.spinOnce();
  delay(100);
}
```

Ros melodic installition

`sudo apt install ros-melodic-desktop`

`sudo apt-get install ros-melodic-rosserial sudo apt-get install ros-melodic-rosserial-arduino`

Roslib installition on Arduino IDE

Navigate to Sketch > Include Library > Manage Library in the IDE and search for `rosserial` package

Rosserial Connection Steps

First of all, we need to run the roscore on terminal. `roscore`

Note: Sometimes permission denied for connecting the related com port error can be seen. To prevent this you can run the chmod command with desired number 1-777 each of them have some specific permission levels. `sudo chmod 666 /dev/ttyUSB0`

On the other hand your computer com port can be named different. USB0-USB1-ACM etc.

Then listen the COM port and listen to data from the board via: `rosrun rosserial_python serial_node.py` ** If any rostopic is created we can use `rostopic pub toggle_led std_msgs/UInt16 "data: 0"` The data can be 0-1 due to our example. For an example of an LED example which is here, we can use this command to trigger the data 1 or 0 via rostopic command.

Summary

In summary, thanks to this project I have learned how to install ros on OS, what is rosserial and how can we use and implement it in our Arduino based boards.

Also, I have understood that rosserial does not work very well on ESP based boards. I recommend using AVR based boards like Arduino Uno, instead of ESPDuino or similar ESP8266 or many different versions of ESP. I have found rosserial has still some open issues regarding this support.

On the other hand, nodemcu boards only supply 3.3v at max. So I have used some USB to TTL converter to simply gain 5V from the computer's USB. Thanks to this converter I have connected my sensor 5v input and servo's 5v input.

The rosserial becomes very popular and easy to use, Arduino programming is also very enjoyable and easy. The rosserial allows us to interface between ros and our development boards, so, enables the use of two very famous and useful technologies to explore new opportunities in robotics. This method allows for distributed computing, centralized control, control abstraction, and several other benefits to robotic systems at a very low cost.

Thanks

Prof. Dr. Haluk Kucuk for supporting me and instructing the lecture.

References

https://medium.com/@mehmetaraksoy/i-want-to-start-with-what-is-not-ros-2245459a9ade https://www.google.com/search?client=firefox-b-d&q=rosserial https://www.ros.org/ https://maker.pro/arduino/tutorial/how-to-use-arduino-with-robot-operating-system-ros https://www.open-electronics.org/presenting-mq-sensors-low-cost-gas-and-pollution-detectors/