

Multi Sensor Robot System (SensorRS), Vehículo robótico multisensorial de exploración controlado por wifi basado en Arduino y Raspberry Pi

Manuel López Urbina



Director: Arturo Morgado Estévez

Universidad de Cádiz
Escuela Superior de Ingeniería

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE
SISTEMAS Y DE LA COMPUTACIÓN

Índice

- 1 Índice
- 2 Objetivos
- 3 Alcance
- 4 Desarrollo
- 5 Temporización
- 6 Herramientas software
- 7 Herramientas hardware
- 8 Raspberry Pi y Arduino

- 9 Complementos hardware
- 10 Sensores
- 11 Esquemático
- 12 Esquemático
- 13 Motores
- 14 Motores
- 15 Comunicaciones
- 16 Conclusiones
- 17 Referencias

Objetivos

SensorRS es un proyecto robótico elaborado principalmente con una placa Raspberry Pi interconectada por puerto serie con una placa Arduino.

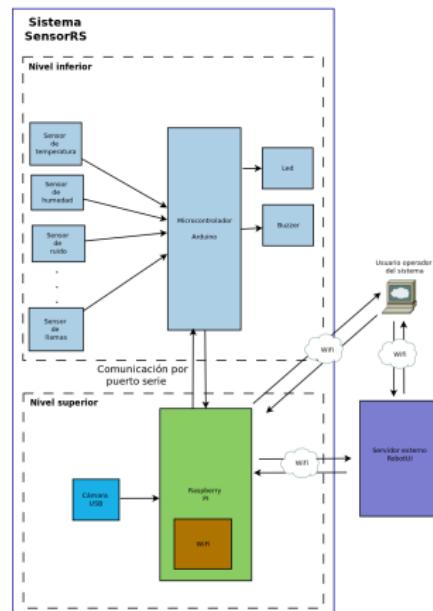
- Sistema dotado de multitud de sensores para captar información del entorno en el que el robot se encuentra.
- Destinado al área de la exploración reduciendo la exposición de personas en zonas peligrosas.
- Sistema basado una arquitectura en microcontroladores programables, mayor escalabilidad, incorporación de más sensores, etc.



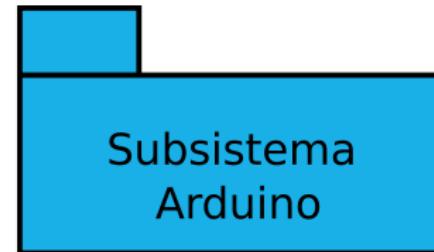
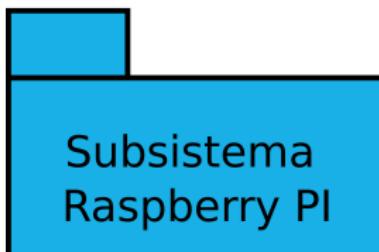
Vehículo SensorRS

Alcance

- Sistema basado en Raspberry Pi gobernada vía wifi a través de la aplicación web RobotUI constituyendo éste el nivel superior del sistema.
- Placa Arduino gestionada por Raspberry Pi encargada del control de sensores.
- Transmisión de vídeo en tiempo real de las imágenes captadas a través de la cámara conectada a la Raspberry Pi.



Subsistemas



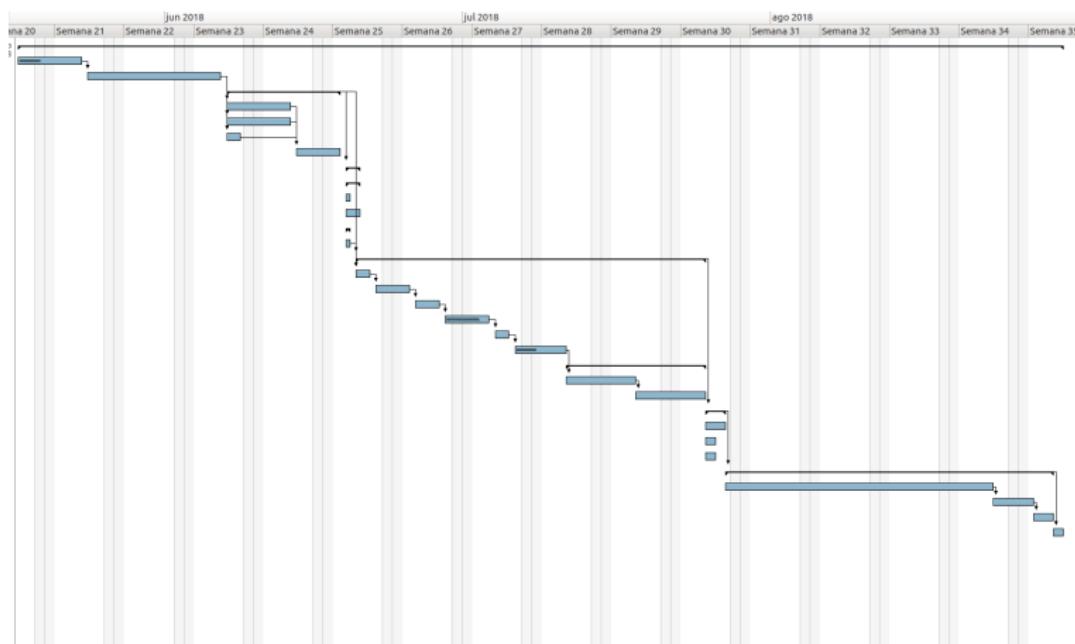
Modelo de ciclo de vida empleado:

- Desarrollo incremental

Temporización

WBS	Nombre	Trabajo
1	▼ SensorRS	89d 3h
1.1	Conocimiento del proyecto	5d
1.2	Planificación y estudio del proyecto	10d
1.3	▼ Análisis de herramientas existentes	15d
1.3.1	Estudio de Node.js	5d
1.3.2	Estudio de Arduino.js	5d
1.3.3	Estudio de Socket.io	2d
1.3.4	Códigos y pruebas	3d
1.4	▼ Definición de requisitos	6d
1.4.1	▼ Requisitos no funcionales	3d
1.4.1.1	Software	1d
1.4.1.2	Hardware	2d
1.4.2	▼ Requisitos funcionales	2d
1.4.2.1	Software	1d
1.4.2.2	Hardware	1d
1.4.3	Reunión de planificación con director del proyecto	1d
1.5	▼ Montaje del vehículo	25d 3h
1.5.1	Instalación y preparación placa Raspberry Pi	2d
1.5.2	Comunicación serie Raspberry Pi - Arduino	2d
1.5.3	Conexión de sensores	3d
1.5.4	Montaje sobre el chasis del vehículo	3d
1.5.5	Alimentación del conjunto	2d
1.5.6	Reunión de seguimiento con director del proyecto	3d 3h
1.5.7	▼ Programación del vehículo	10d
1.5.7.1	Placa Raspberry Pi	5d
1.5.7.2	Programación Arduino	5d
1.6	▼ Mejoras en la aplicación RobotUI	4d
1.6.1	Uso del gamepad	2d
1.6.2	Renovación de la interfaz	1d
1.6.3	Mejoras en comunicaciones	1d
1.7	▼ Documentación	23d
1.7.1	Memoria	19d
1.7.2	Resumen	2d
1.7.3	Presentación	2d
1.8	Reunión de seguimiento con el director del proyecto	1d

Temporización





Arduino IDE

Placas utilizadas

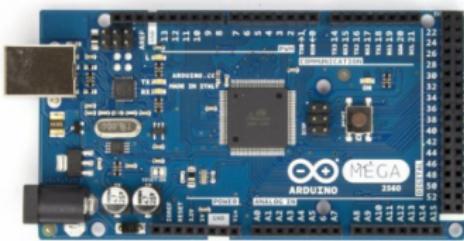
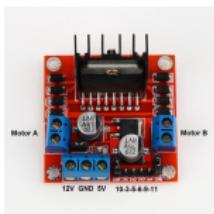
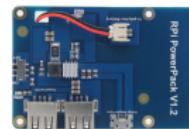


Imagen de una Raspberry Pi (izquierda) y una Arduino Mega (derecha) utilizadas en SensorRS.

Complementos hardware



Driver de motores L298N.



Módulo de alimentación para Raspberry Pi.



Alimentador USB para protoboard.

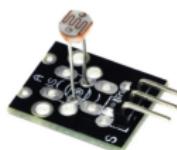


Cámara USB alta definición.

Sensores utilizados



Sensor DHT11



Fotoresistor



Sensor KY-038



Sensor YG1006



Sensor YG1006



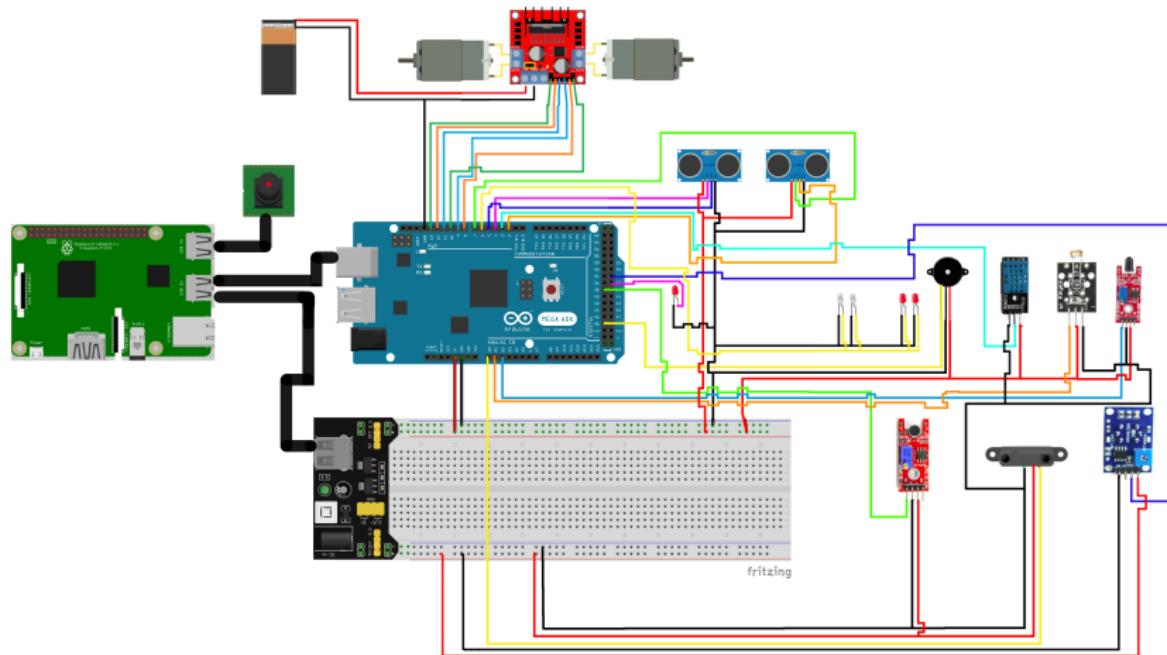
Sensor MQ-2



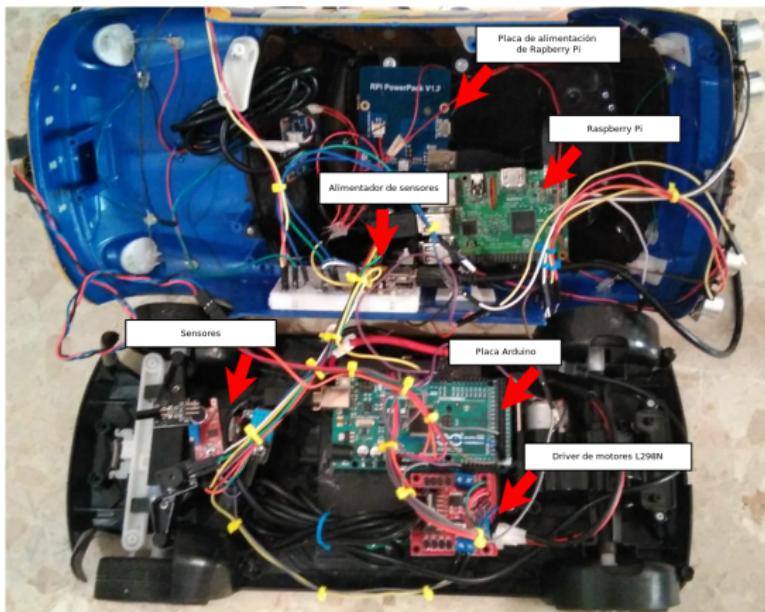
SHARP GP2D12

Conjunto de sensores integrados en SensorRS.

Conexionado general

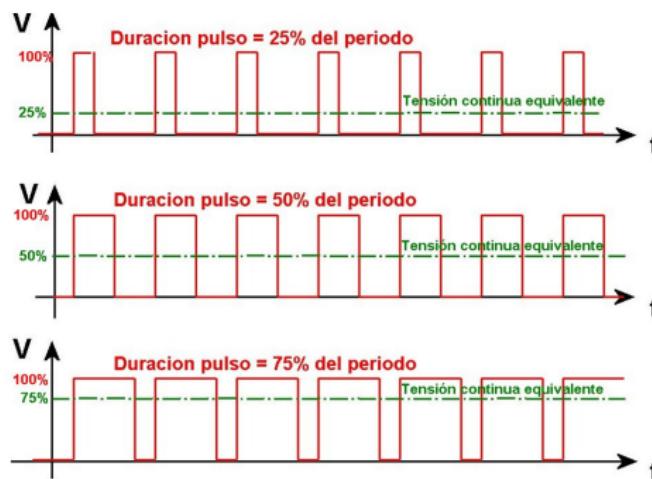


Conexionado general



Esquemático del conexionado general del robot.

PWM

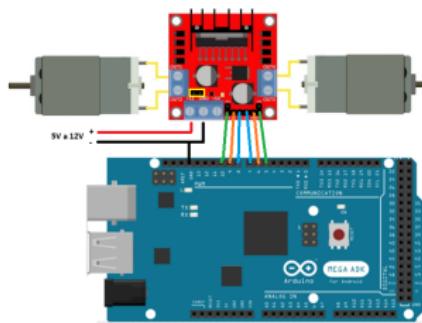


Señales PWM.

Control de motores



Motor de dirección.



Conexionado.



Módulo de tracción.

Comunicaciones Serie

Transmisión de Raspberry Pi a Arduino:

```
const SerialPort = require ('serialport');
const Readline = SerialPort.parsers.Readline;
const port = new SerialPort('/dev/ttyACM0',{ baudRate: 9600 });
const parser = new Readline();
port.pipe(parser);

function serial_transmission(data ,delay){
    setTimeout(function(){
        //Env o del string caracter por caracter
        for(vari=0;i<data.length ;i++){
            port.write(newBuffer(data[i] , 'ascii '));
        }

        //Caracter final \n
        port.write(newBuffer( '\n' , 'ascii '));
    },delay );
}
```

Comunicaciones Serie

Captura de mensaje en Raspberry Pi procedente de Arduino:

```
const parser = port.pipe(new Readline({ delimiter: '\r\n'}));  
parser.on('data', function(data){  
    console.log(data) ;  
    var msg_data = data.split("%");  
    socket.emit(String(msg_data[0]), { msg: String(msg_data[1]) })  
}) ;
```

Siendo msg_data[0] el indicativo del tipo de dato y msg_data[1] su valor

Comunicaciones Serie

Captura de mensaje en Arduino procedente de Raspberry Pi:

```
void loop(){
    while(Serial.available()>0){
        charreceived=Serial.read();
        inData.concat(received);
        if(received==\n){
            inData.trim();
            //MOTOR
            if(inData.startsWith("MOTOR")){
                if(sscanf(inData.c_str(),"MOTOR-%d",&motor)==1){
                    Serial.println(motor);
                }
                if(motor<=51 && motor >= -51 ){
                    Serial.println('STOP');
                }
            }
            inData="";
        }
        delay(100);
    }
}
```

Comunicaciones Serie

Envío de mensaje a Raspberry Pi procedente de Arduino:

```
void loop(){
    byte temperature=0;
    byte humidity=0;
    int err=SimpleDHTErrSuccess;
    if((err=dht11.read(pinDHT11,&temperature,&humidity, NULL))
        Serial.print("ReadDHT11failed,err=");
        Serial.println(err);delay(1000);
    }
    dht11.read(pinDHT11,&temperature,&humidity, NULL);

    if((int)temperature!=0and(int)humidity!=0){
        Serial.print("tmp%");
        Serial.print((int)temperature);Serial.print("*C,");
        Serial.print((int)humidity);Serial.println("H");
    }

    delay(30);
    Serial.println(analogRead(photosensorPin));
}
```

Conclusiones

La realización del trabajo *SensorRS: Asistente de diseño de interfaz de control, seguimiento y sharing de robots en tiempo real* como trabajo de fin de carrera, se ha caracterizado por:

- La elaboración de un vehículo robótico haciendo uso de una Raspberry Pi 3 Model B.
- Programación del microcontrolador Atmega2560 alojado en una placa Arduino.
- Comunicación por puerto serie entre la placa Arduino y Raspberry Pi
- Realización de conexiones entre elementos y montaje del vehículo.
- Transmisión de gran cantidad de datos entre cliente servidor y servidor cliente. Streaming de vídeo y audio, emisión de comandos entre otros datos.

-  Hau-Shiue Juang and Kai-Yew Lum.
Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller Board.
Technical report, IEEE International Conference on Control and Automation, 5 2013
-  Kimmo Karvinen Tero Karvinen.
Make: Getting Started with Sensors.
Make Media Books, 2014.
-  Nilson Mori Lazarin and Carlos Eduardo Pantoja.
A Robotic-agent Platform For Embedding Software Agents usin Raspberry Pi and Arduino Boards.
Technical report, UnED Nova Friburgo, 6 2011.
-  Leonel G. Corona Ramírez Griselda S. Abarca Jiménez Jesús Mares Carreño.
Sensores y actuadores, Aplicaciones con Arduino.
Grupo Editorial Patria, 2014.}
-  D. Grahame Holmes Thomas A. Lipo.
Pulse Width Modulation for Power Converters.
Power Engineering, 2012.
-  Lakshminarasimhan Srinivasan, Julian Scharnagl, and Klaus Schilling.

Analysis of WebSockets as the New Age Protocol for Remote Robot Tele-operation.

Technical report, University of Wuerzburg, Department of Robotics and Telematics, 11 2013.

-  Lakshminarasimhan Srinivasan, Julian Scharnagl, Zhihao Xu, Nicolas Faerber, Dinesh K. Babu, and Klaus Schilling.
Design and Development of a Robotic Teleoperation System using Duplex WebSockets suitable for Variable Bandwidth Networks.
Technical report, University of Wuerzburg, Department of Robotics and Telematics, 11 2013.
-  Nazirah Ahmad Zaini, Norliza Zaini, Mohd Fuad Abdul Latip, and Nabilah Hamzah.
Remote Monitoring System based on a Wi-Fi Controlled Car Using Raspberry Pi.
Technical report, Universiti Teknologi MARA (UiTM) Shah Alam, Malaysia, Faculty of Electrical Engineering, 12 2016.

Multi Sensor Robot System (SensorRS), Vehículo robótico multisensorial de exploración controlado por wifi basado en Arduino y Raspberry Pi

Manuel López Urbina



Director: Arturo Morgado Estévez

Universidad de Cádiz
Escuela Superior de Ingeniería

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE
SISTEMAS Y DE LA COMPUTACIÓN