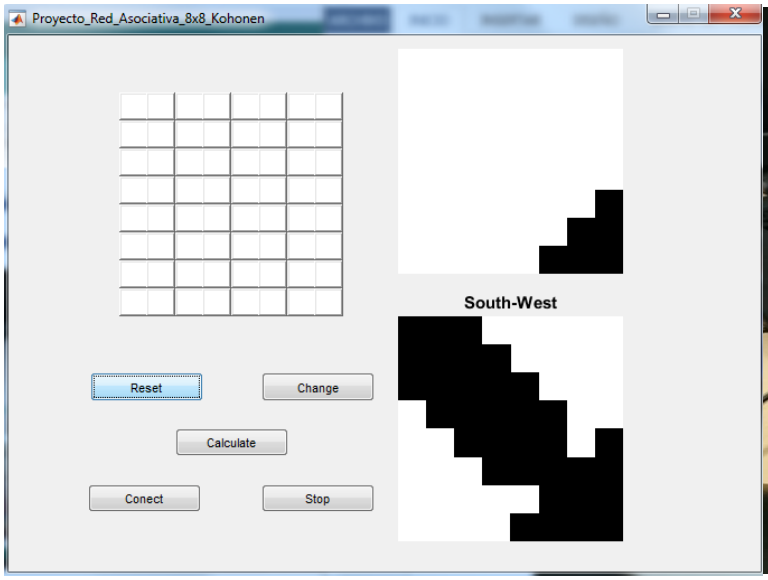
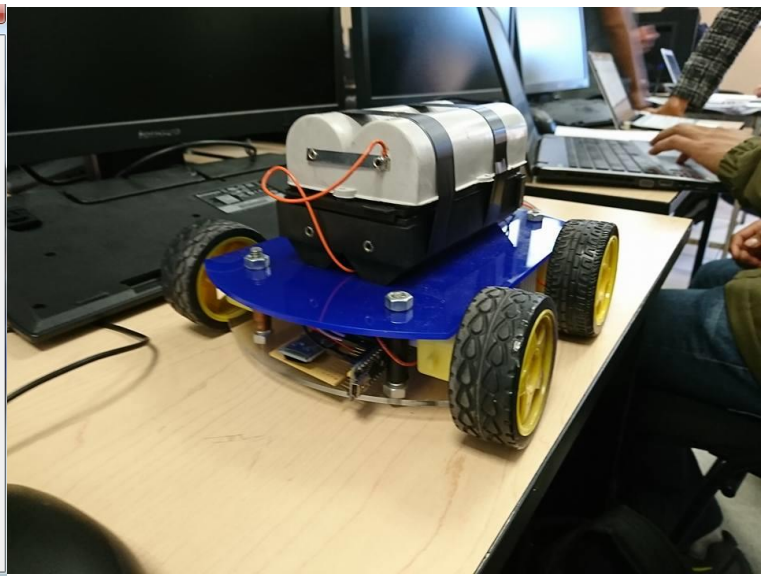
Recognizing of the way a Bluetooth Controlled car must follow through the self-organizing Kohonen Neural Network.





García Castán Juan Carlos

Zarazua Aguilar Luis Fernando.

INDEX

1. Introduction

Man has been looking for simplify each process in his life by building machines or developing methods that could solve those problems. Neural Networks are part of that development and they can give us a lot of solutions for different problems in different areas.

Neural Networks can emulate some characteristics of humans as the way as humans learn and associate stuff, that is to say, they can get experience through repeating methods that have something in common. In short, neural networks are just an artificial and simplified model of the brain which is the most perfect system to describe this process, the steps of the process happen around a neuron are basically the stimulation of a neuron (input), then, when the input reach some threshold, it gets activate and sends a signal towards the axon.

As far as we know thinking goes from a neuron to another, and to do that possible brain has a lot of interconnected neurons, then, the secret of intelligence is the interconnection of those neurons and the interaction between them, also we know that humans can learn and that means that those problems we didn't know how to solve could being solved after getting information about them.

So Neural Networks:

- Are process units which exchange data or information
- Are useful to recognize patterns, images, time etc.
- Can learn and improve.

In this project; the user must draw the direction by touching some squares in a MATLAB® GUI then a Bluetooth controlled car will follow that direction. This kind of control lets a user draw a shaft as he considers it's correct and the neuron, from the information it has, starts to linking the draw to the known figures of the directions, even the neuron can learn about other symbols that mean other things, for example to stop the car or even, using the right elements, increasing or decreasing the how fast or slot the car must go.

We performed 9 classes which describe North, South, East, West, North-East, North-West, South-East, South-West directions and 'Stop'.

To divide this classes, we used 9 patterns for each class, so, when a user does drawing in the GUI the neurons start recognizing the kind of figure trough the areas selected and then it compares the figure with the patterns the neural network learned and send a command to the car to move it to the direction given or to stop moving.

We implemented an Kohonen Neural Network to identify the direction the final user chooses because of the advantages of Kohonen Neural Network over Adaline or Perceptron, that we think are:

- The neurons that doesn't learn disappear.
- The adaptation to formerly unknown input data.[1]
- Self-organization.

This kind of self- organization must figure out the common traits regularities, correlations or categories at the input data, the selection of the 'best neuron' allows the elimination of the others forcing them to give their minimum response values.

So, the objective of the neuron is categorizing the input data in those that are similar in a singular class and then the values at the input that are similar to the others, that were self-organizing by the neuron, goes to that class.

With all the advantages of this neural network, we made a neuron which learned about 9 kinds of figures, then it started the auto classification of each image in 9 classes, as we can see in the MATLAB code at the annex of this document.

The first step of our process is select a vector to calculate the distance between the vector and the 'w', after obtain the closest vector best matching unit (BMU) the other ones are update, and the BMU goes towards the vector. The value of the attraction between them is ruled by the learning rate.[2]

It's very important that evenness doesn't occur. When the Neural network is learning the w's are slight random values and they are changing for being the best election of the input data.

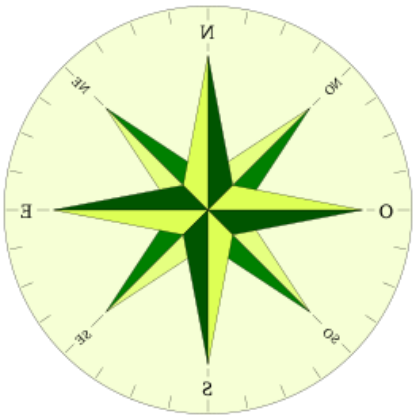
After the training the neural map gets reorganized, adapting the wages values of the winning neuron and his neighbors according to the response. We can see the learning algorithm in figure 1:

$$m_j(t+1) = \begin{cases} m_j(t) + \alpha(t)\,(x(t) - m_j(t)) & j \in N_c(t) \\ m_j(t) & j \notin N_c(t) \end{cases}$$

figure 1: Kohonen Algorithm.

2. Objective

Apply A Kohonen Neural Network which help us to identify, through a drawing created by a user, the direction a car, controlled by Bluetooth, must follow.



The directions are shown in figure 2. The user can also decide if the car doesn't have to move.

Figure 2: directions

3. Procedure

First of all, we designed the graphical interface which has an 8x8 array, it is inside a GUI and there we can select the squares that will draw the image, at the code we initialized the neuron with the 4 patterns of each class.

As the Kohonen method says, we obtained random values for 'w' and started the classification, for being more accurate we chose 800 epochs, this information we can see it at the code which is in the annex part of the document. After classifying the classes, we started to evaluate what the user introduced in the GUI , when the user select an square, that is a 'togglebutton', it changes its color, this process is very similar to the practice 6 process in there we had to draw an image and following that we used the way of creating the image and filling the squares. After the user select the squares needed, he must press the 'Calculate' button as we can see in figure 3, then the program starts to calculate, from the squares selected, the value of 'a' that tell us the class the image belongs.

In the code we use the switch-case statement after calculating the value of 'a', and we send a previous created image to the GUI and the direction the car will take.

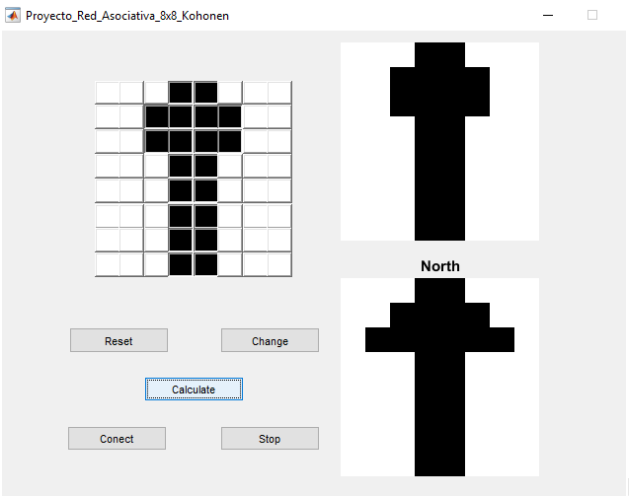


Figure 3: User interface.

If the image is not recognized, the program sends an error, and the car won't move.

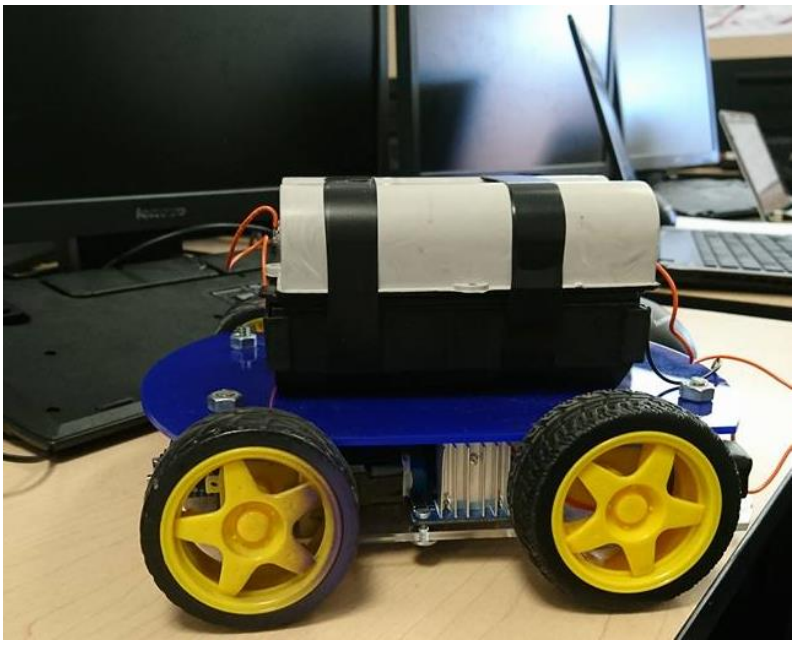


figure 4: Car utilized.

When the user wants to delate the image he drew, he must press the reset button and all the squares will make white again. The button ‘Change’ change automatically the white blocks to black and vice versa.

The program works with and without the Bluetooth connection, if the user wants to move the car he has to press the connect button and the interface will make the communication between the program and the car and it will move. Finally the ‘Stop’ button stops the Bluetooth communication, and all the process, that is to say, the car won’t move after that.

For the Bluetooth communication we implemented an ‘Arduino nano’. The arduino code is in the Annex part of the document, and the connection was set in the MATLAB® Code.

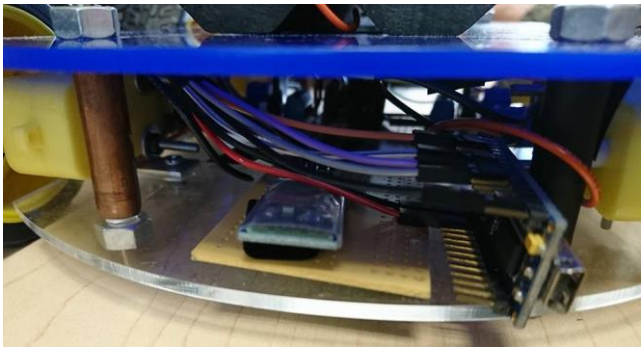


figure 5: Arduino nano, utilized in the car.

4. Results

The program can recognize the patterns at the input very well, it doesn't matter if the figure is not the same as the previous introduced patterns for the training (figure 6). We realized that the area which has more filled squares will be the direction the neural network sends to the car, an example of this characteristic is in figure 6.

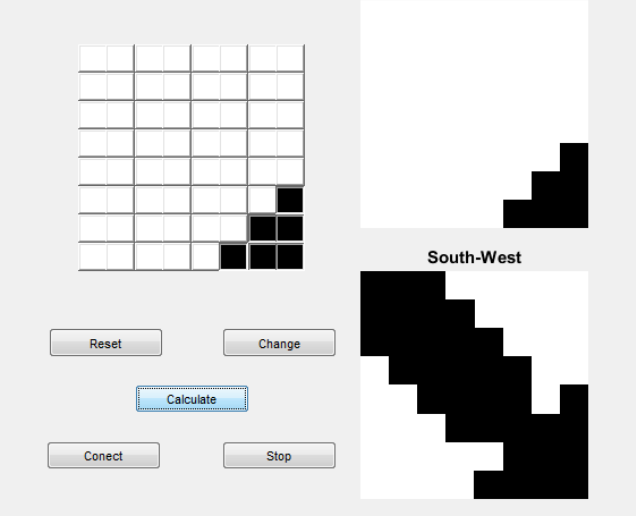


figure 6. How the program identify the figure.

That behavior pattern is very useful because each user can draw the arrow as he thinks it is properly and the neural network will recognize what he wants.

In figure 6 are the patterns that the Neural Network learned.

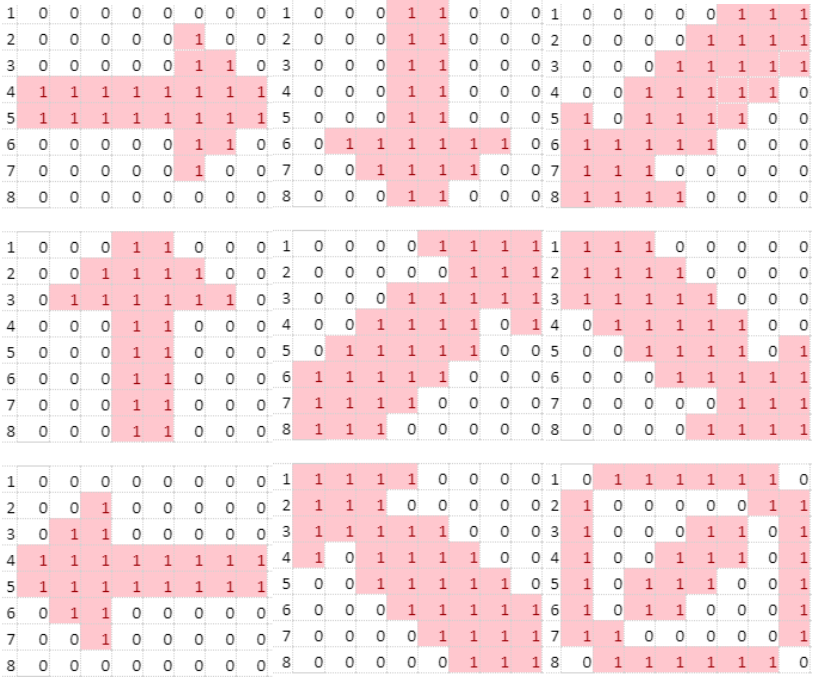


figure 7: Patterns of the NN

5. Conclusions

- **García Castán Juan Carlos:**

The Kohonen Neural Network is one of the most closer to the brain work artificial neurons, it's very important the way it can rich the self-organization because it lets itself learn about new stuff. This application is just a little part of all the applications it could take. Neural Networks are improving every day, and we can see this kind of technology in our daily life.

The capability of this neural networks let us expand the things we can make over this project, we can control more than the direction, for example the velocity of the car, and it shows how powerful this neural network could be.

- **Zarazua Aguilar Luis Fernando:**

The kohonen neural network proved to be adequate for the process of recognizing images for the direction of a cart, it because the patterns had a very easy to notice feature and it was that at be arrows there were a large number of pixels were concentrated in one of the edges, thus delimiting the patterns in sets by their similarities, however in the training if the initial weights were not chosen correctly we will have an incorrect result due there was not known in which set each weight fell and in some cases it was not possible identify it correctly giving an incorrect direction.

6. References

[1] http://galaxy.agh.edu.pl/~vlsi/AI/koho_t/index_eng.html
[2] http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema5dm.pdf
[3] http://www.unilibre.edu.co/revistaavances/avances_9/r9_art2.pdf

https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf

7. Annex

7.1 Arduino Code:

```
const int Lampara = 13;

const int LLanta_Izq_A=5;//Pin de la llanta Izquierda_A.

const int LLanta_Izq_B=6;//Pin de la llanta Izquierda_B.

const int LLanta_Der_A=3;//Pin de la llanta Derecha_A.

const int LLanta_Der_B=11;//Pin de la llanta Derecha_B.

const int Led_Prueba=9;//Pin del Led de Prueba.
```

```
const int Paro=0;//Motor en Paro.
const int Adelante=1;//Motor Girando hacia adelante.
const int Atras=2;//Motor Girando hacia átras.
const int Vel_1=1;//Velocidad Moderada.
const int Vel_2=2;//Velocidad Rapida.
int Carga_PWM,Carga_PWM2,Opcion_A;
void setup()
{
  // initialize the serial communication:
  Serial.begin(9600);
  // initialize the ledPin as an output:
  pinMode(Lampara, OUTPUT);
  pinMode(LLanta_Izq_A, OUTPUT);
  pinMode(LLanta_Izq_B, OUTPUT);
  pinMode(LLanta_Der_A, OUTPUT);
  pinMode(LLanta_Der_B, OUTPUT);
  pinMode(Led_Prueba, OUTPUT);
  TCCR0A =163;//1010,0011 Modo PWM.
  TCCR0B =03;//0000,0011 factor de 64
  TCCR2A =163;
  TCCR2B =03;
  Carga_PWM=200;
  Carga_PWM2=100;
  Opcion_A=8;
  Motor(Paro,Vel_1,Paro,Vel_1);
}
void loop() {
int Opcion,Carga;
 if (Serial.available()>0) {
   Opcion = Serial.parseInt();
   if (Serial.read() == 'r') {
    //Serial.println(Opcion);
    if(Opcion == 0){//Giro Este.
     Motor(Adelante,Vel_2,Paro,Vel_1);}
    else if(Opcion == 1){//Avance NorEste.
```

```
  Motor(Adelante,Vel_2,Adelante,Vel_1);}
 else if(Opcion == 2){//Avance Norte.
  Motor(Adelante,Vel_2,Adelante,Vel_2);}
 else if(Opcion == 3){//Avance NorOeste.
  Motor(Adelante,Vel_1,Adelante,Vel_2);}
 else if(Opcion == 4){//Giro Oeste.
  Motor(Paro,Vel_1,Adelante,Vel_2);}
else if(Opcion == 5){//Avance SurOeste.
  Motor(Atras,Vel_1,Atras,Vel_2);}
else if(Opcion == 6){//Avance Sur.
  Motor(Atras,Vel_2,Atras,Vel_2);}
else if(Opcion == 7){//Avance SurEste.
  Motor(Atras,Vel_2,Atras,Vel_1);}
else if(Opcion == 8){//Paro del Vehiculo.
  Motor(Paro,Vel_1,Paro,Vel_1);}
else if(Opcion == 9){//Prender Lámpara.
  digitalWrite(Lampara,HIGH);}
else if(Opcion == 10){//Apagar Lámpara.
  digitalWrite(Lampara,LOW);}
else if(Opcion == 11){//Ajustar Velocidad.
  Carga=Serial.parseInt();
  if (Serial.read() == 'r'){
  Carga_PWM=Carga;
  Carga_PWM2=Carga/2;
  //Motor(Paro,Vel_1,Paro,Vel_1);
  //Modificar Velocidad Adecuadamente
  if(Opcion_A == 0){//Giro Este.
  Motor(Adelante,Vel_2,Paro,Vel_1);}
 else if(Opcion_A == 1){//Avance NorEste.
  Motor(Adelante,Vel_2,Adelante,Vel_1);}
 else if(Opcion_A == 2){//Avance Norte.
  Motor(Adelante,Vel_2,Adelante,Vel_2);}
 else if(Opcion_A == 3){//Avance NorOeste.
  Motor(Adelante,Vel_1,Adelante,Vel_2);}
 else if(Opcion_A == 4){//Giro Oeste.
```

```
    Motor(Paro,Vel_1,Adelante,Vel_2);}
   else if(Opcion_A == 5){//Avance SurOeste.
    Motor(Atras,Vel_1,Atras,Vel_2);}
   else if(Opcion_A == 6){//Avance Sur.
    Motor(Atras,Vel_2,Atras,Vel_2);}
   else if(Opcion_A == 7){//Avance SurEste.
    Motor(Atras,Vel_2,Atras,Vel_1);}
   else if(Opcion_A == 8){//Paro del Vehiculo.
    Motor(Paro,Vel_1,Paro,Vel_1);}
    ///Fin de elección*/
    analogWrite(9,Carga_PWM);} }
   if(Opcion<9){
   Opcion_A=Opcion;}
   Serial.flush();
  }
 }
}

 void Motor(int Motor_Izq,int Vel_Izq,int Motor_Der,int Vel_Der){//Llanta Izquierda(xx),,Llanta Derecha(xx)
  //Motor Izquierdo.
  if(Motor_Izq==Paro){
  analogWrite(LLanta_Izq_A,0);
  analogWrite(LLanta_Izq_B,0);}
  //
  else if(Motor_Izq==Adelante && Vel_Izq==Vel_1){
  analogWrite(LLanta_Izq_A,Carga_PWM2);
  analogWrite(LLanta_Izq_B,0);}
  //
  else if(Motor_Izq==Adelante && Vel_Izq==Vel_2){
  analogWrite(LLanta_Izq_A,Carga_PWM);
  analogWrite(LLanta_Izq_B,0);}
  //
  else if(Motor_Izq==Atras && Vel_Izq==Vel_1){
  analogWrite(LLanta_Izq_B,Carga_PWM2);
  analogWrite(LLanta_Izq_A,0);}
```

```
 //
 else if(Motor_Izq==Atras && Vel_Izq==Vel_2){
 analogWrite(LLanta_Izq_B,Carga_PWM);
 analogWrite(LLanta_Izq_A,0);}
//Motor Derecho
 if(Motor_Der==Paro){
 analogWrite(LLanta_Der_A,0);
 analogWrite(LLanta_Der_B,0);}
 //
 else if(Motor_Der==Adelante && Vel_Der==Vel_1){
 analogWrite(LLanta_Der_A,Carga_PWM2);
 analogWrite(LLanta_Der_B,0);}
 else if(Motor_Der==Adelante && Vel_Der==Vel_2){
 analogWrite(LLanta_Der_A,Carga_PWM);
 analogWrite(LLanta_Der_B,0);}
 else if(Motor_Der==Atras && Vel_Der==Vel_1){
 analogWrite(LLanta_Der_B,Carga_PWM2);
 analogWrite(LLanta_Der_A,0);}
 else if(Motor_Der==Atras && Vel_Der==Vel_2){
 analogWrite(LLanta_Der_B,Carga_PWM);
 analogWrite(LLanta_Der_A,0);}
}
```

7.2 Matlab Code: (Because of the length of the code, we just added the Kohonen and the communication part.)

```
    %%Kohonen Method

function Inicializar_Neurona()
global W P P1a P2a P3a P4a P5a P6a P7a P8a P9a
P1a=[0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0];
P2a=[0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0];
P3a=[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0];
P4a=[0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0];
P5a=[0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,1,1,1,0,0,1,1,1,1,0,1,0,1,1,1,1,1,1,0,0,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0];
P6a=[1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,1,0,1,1,1,1,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1];
P7a=[0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,0,0,1,1,1,1,1,1,0,1,0,1,1,1,1,1,0,0,1,1,1,1,1,0,0,0,0,1,1,1,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,0];
P8a=[1,1,1,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,1,1,1,1,1,1,0,1,0,0,0,1,1,1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,1];
P9a=[0,1,1,1,1,1,1,0,1,0,0,0,0,0,0,1,1,1,0,0,0,1,1,0,1,1,0,0,1,1,1,0,1,1,0,1,0,1,1,1,0,0,1,1,0,1,1,0,0,0,0,1,1,1,0,0,0,0,0,1,0,1,1,1,1,1,1,0];
%%%%
P1b=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0];
P2b=[0,0,0,1,1,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0];
P3b=[0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
P4b=[0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,0,0,0,0,1,1,0,0,0,0];
P5b=[0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,1,1,1,0,0,1,1,1,1,0,1,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0];
P6b=[1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,1,0,1,1,1,1,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,1,1];
```

```
P7b=[0,0,0,0,0,0,1,1,0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,1,0,1,1,1,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0];
P8b=[1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,1,0,1,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1];
P9b=[0,1,1,1,1,1,1,0,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,1,0,1,1,0,0,0,1,1,0,1,1,0,0,1,1,0,0,1,0,1,1,0,0,0,1,1,1,0,0,0,0,0,0,1,0,1,1,1,1,1,1,0];
%%%%
P1c=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0];
P2c=[0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0];
P3c=[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0];
P4c=[0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0];
P5c=[0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,1,0,1,0,0,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0];
P6c=[1,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1];
P7c=[0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,1,0,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0];
P8c=[1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,1,1,1];
P9c=[0,1,1,1,1,1,1,0,1,0,0,0,0,0,0,0,1,1,0,0,0,1,1,0,1,1,0,0,1,1,0,0,1,1,0,1,1,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,0,1,1,1,1,1,1,0];
%%%%
P1d=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
P2d=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0];
P3d=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
P4d=[0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
P5d=[0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,1,0,0,0,0,0,1,0,0,1,0,1,0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0];
P6d=[1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1];
P7d=[0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,0,0,1,0,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0];
P8d=[1,1,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,1,1,1];
P9d=[1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,1,0,1,1,0,0,0,0,1,0,0,1,1,0,0,1,0,0,0,1,1,0,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1];
P=[P1a' P1b' P1c' P1d',...
   P2a' P2b' P2c' P2d',...
   P3a' P3b' P3c' P3d',...
   P4a' P4b' P4c' P4d',...
   P5a' P5b' P5c' P5d',...
   P6a' P6b' P6c' P6d',...
   P7a' P7b' P7c' P7d',...
   P8a' P8b' P8c' P8d',...
   P9a' P9b' P9c' P9d'];
[NE,NP]=size(P);
NF=9;
alpha=0.2;
n=0.1;
w1=P1a+rand(1,NE)*n;
w2=P2a+rand(1,NE)*n;
w3=P3a+rand(1,NE)*n;
w4=P4a+rand(1,NE)*n;
w5=P5d+rand(1,NE)*n;
w6=P6b+rand(1,NE)*n;
w7=P7a+rand(1,NE)*n;
w8=P8a+rand(1,NE)*n;
w9=P9a+rand(1,NE)*n;
edist(NF)=0;
W=[w1;w2;w3;w4;w5;w6;w7;w8;w9];
for epochs=1:800
 for n=1:NP
  for i=1:NF
   edist(i)=Norma(W(i,:),P(:,n));%%sqrt((W(i,1)-P(1,n))^2+(W(i,2)-P(2,n))^2);
  end
  Winner=compet((-edist)');
  for z=1:NF
   W(z,:)=W(z,:)+alpha*Winner(z)*(P(:,n)'-W(z,:));
  end
 end
end
W(5,:)=P5b*0.95;
W(6,:)=P6a;
R(NP)=0;
for j=1:NP
 po=P(:,j)';
 a=compet(W*po');
 r=0;
 for i=1:length(a)
```

```
    r=r+a(i)*i;
  end
  R(j)=r
end


        %%Comunication:

function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
global W P1a P2a P3a P4a P5a P6a P7a P8a P9a conexion puerto
po=Obtener_Valores(handles);
a=compet(W*po');
axes(handles.axes1),imshow(Create_Image(po));
axes(handles.axes2)
r=0;
for i=1:length(a)
    r=r+a(i)*i;
end
% axes(handles.axes2),imshow(Create_Image(a));
switch r
    case 1
        imshow(Create_Image(P1a));
        title('Este');
        if conexion
          fprintf(puerto,'0r');
        end
    case 2
        imshow(Create_Image(P2a));
        title('Norte');
        if conexion
          fprintf(puerto,'2r');
        end
    case 3
        imshow(Create_Image(P3a));
        title('West');
        if conexion
          fprintf(puerto,'4r');
        end
    case 4
        imshow(Create_Image(P4a));
        title('Sur');
        if conexion
          fprintf(puerto,'6r');
        end
    case 5
        imshow(Create_Image(P5a));
        title(North-West');
        if conexion
          fprintf(puerto,'1r');
        end
    case 6
        imshow(Create_Image(P6a));
        title('North-East');
        if conexion
          fprintf(puerto,'3r');
        end
    case 7
        imshow(Create_Image(P7a));
        title('South-East');
        if conexion
          fprintf(puerto,'5r');
        end
    case 8
        imshow(Create_Image(P8a));
        title('South-West');
```

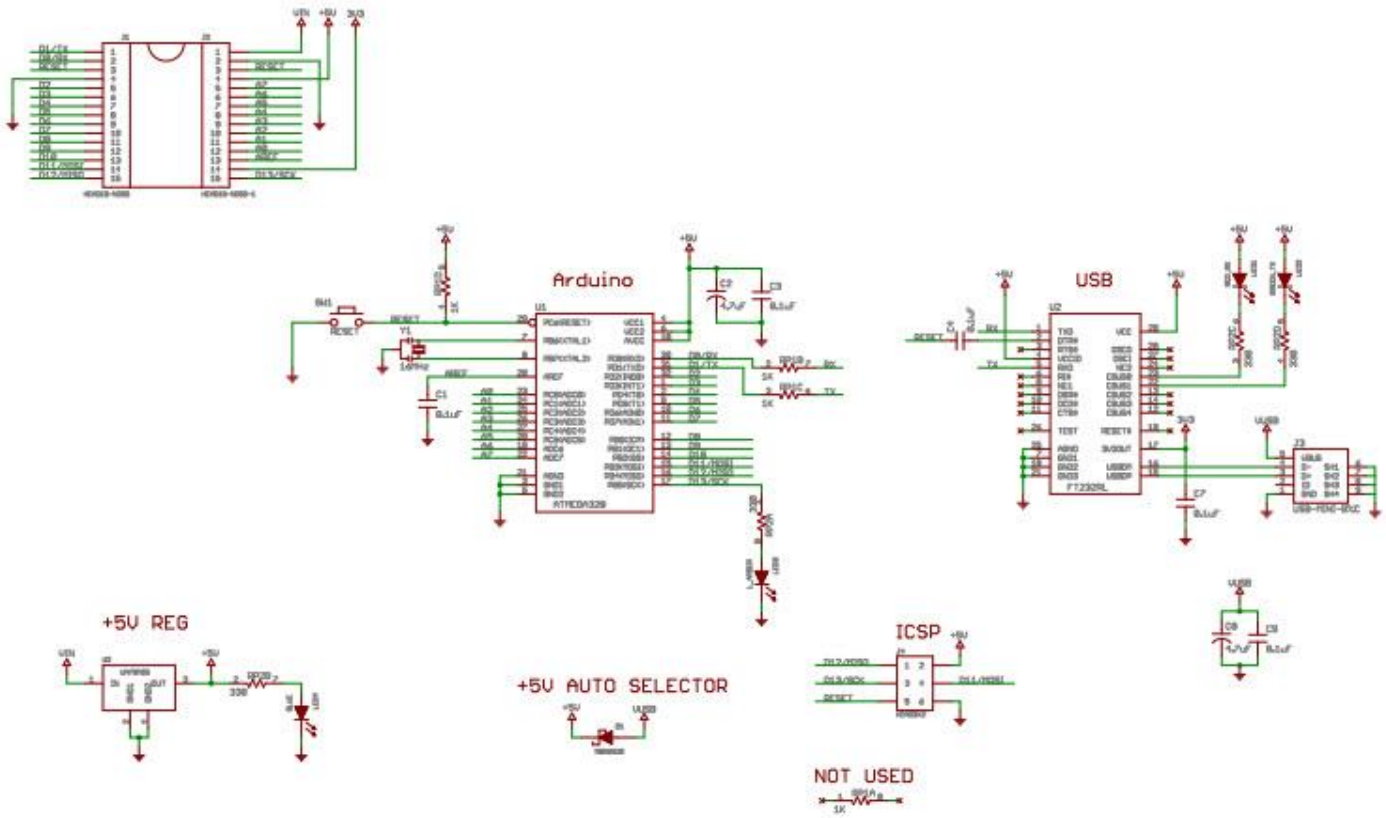
```
        if conexion
          fprintf(puerto,'7r');
        end
    case 9
        imshow(Create_Image(P9a));
        title('Paro');
        if conexion
          fprintf(puerto,'8r');
        end
    otherwise
        imshow(Create_Image(P9a));
        title('Desconocido');
        if conexion
          fprintf(puerto,'8r');
        end
end


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
global puerto conexion
delete(instrfind({'Bluetooth'},{'HC-06'}));
puerto=Bluetooth('HC-06',1);
% delete(instrfind({'port'},{'COM5'}));
% puerto=serial('COM5');
% puerto.BaudRate=9600;
try
   fopen(puerto);%abre el puerto a utilizar
   disp('Conexion correcta')
   conexion=true;
%    fprintf(puerto,'Hola');
%    fclose(puerto);
catch
   delete(puerto);
   disp('Conexion Erronea')
   conexion=false;
end

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
global puerto conexion
if conexion
     fprintf(puerto,'8r');
end
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

7.3 Arduino nano schematic



7.4 HC-06 Bluetooth to serial Module:

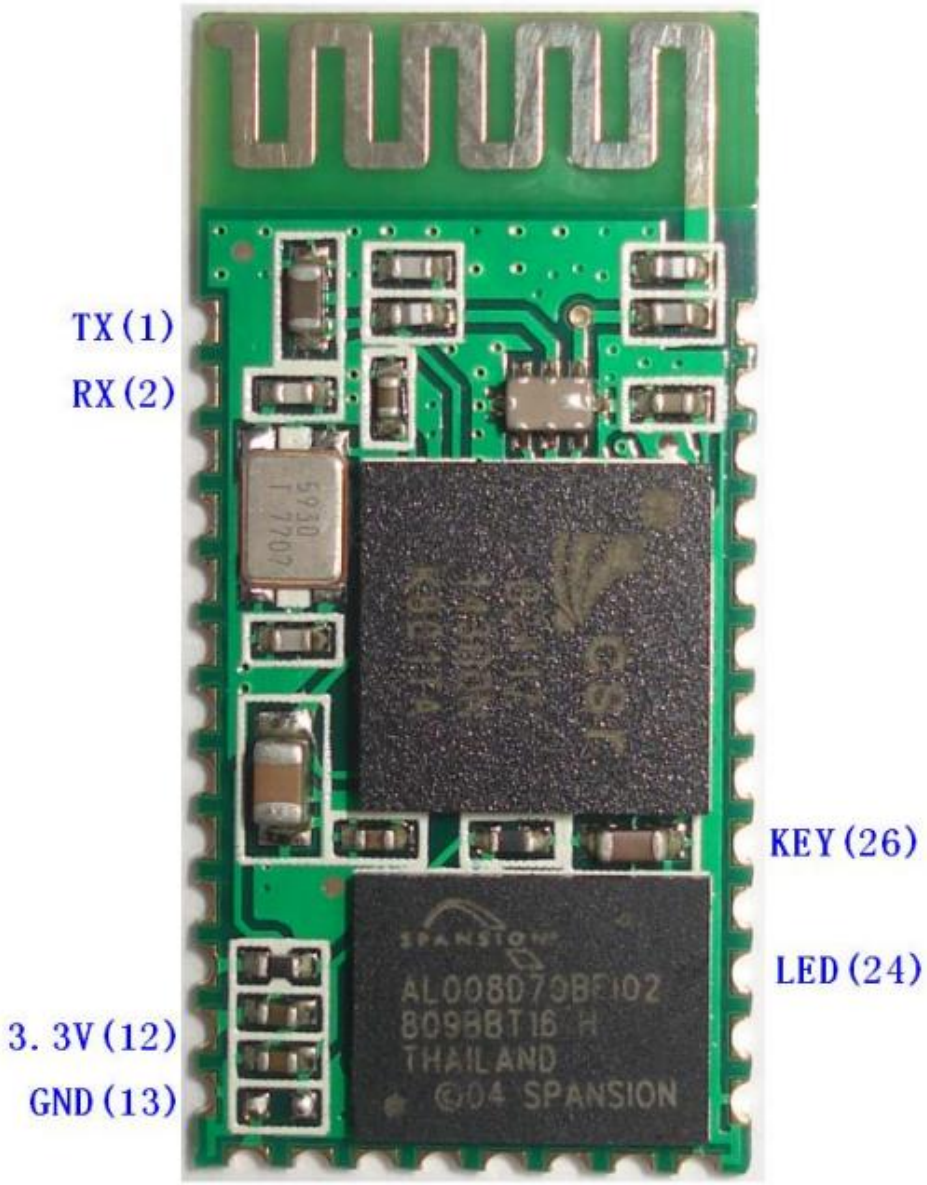


Figure 1 A Bluetooth module