



Universidad Veracruzana

UNIVERSIDAD VERACRUZANA

FACULTAD DE INSTRUMENTACIÓN
ELECTRÓNICA



FACULTAD
DE
INSTRUMENTACIÓN
ELECTRÓNICA

5-11-2020

TÓPICOS AVANZADOS DE INSTRUMENTACIÓN ELECTRÓNICA I (SISTEMAS EMBEBIDOS)

Tarea 7. – Comunicación

Alumnos:

Rangel Pulido Julio

Martínez González Sergio David

Ramírez Mendoza Jesús Emiliano

Catedrático:

Hernández Reyes Sergio Machuca

ÍNDICE

1.	ACTIVIDAD	3
2.	DESCRIPCIÓN	3
3.	LISTA DE COMPONENTES	6
4.	DIAGRAMA.....	7
5.	CONCLUSIONES	8
6.	BIBLIOGRAFÍA	9

1. ACTIVIDAD

Realizar una aplicación manipulada por un usuario(pc) y un servicio(arduino) esta aplicación debe ejercitar la comunicación bidireccional entre la plataforma y la computadora

2. DESCRIPCIÓN

Sensor de proximidad y encendido de un led:

Para esta actividad es necesario llevar a cabo comunicación bidireccional para esto se usará una tarjeta arduino Mega (puede ser cualquier versión que se desee) con su respectiva id de arduino para programarlo, en la computadora se usará processing para crear la UI con la cual se va a interactuar con la placa y el medio por el cual se llevará a cabo esta interacción será con el puerto serial.

En el caso del control del led será necesario crear un par de botones en processing uno para encendido y uno para apagado, para poder visualizar la información del sensor se creará una animación con la cual se pueda apreciar los valores recibidos, así como letreros en los cuales se muestren los valores numéricos del mismo.

A continuación, se muestra el primer código el cual será usado en la tarjeta arduino.

Líneas del código Arduino(C++).

```
const int Trigger = 53;    //Pin digital 2 para el Trigger del sensor
const int Echo = 52;      //Pin digital 3 para el echo del sensor
int led = 51;

int t = 0;
int d = 0;

void setup() {
    Serial.begin(9600); //inicializamos la comunicación
    pinMode(Trigger, OUTPUT); //pin como salida
    pinMode(Echo, INPUT); //pin como entrada
    digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
    pinMode(led, OUTPUT);
}

void loop() {
    if (Serial.available()) {
        char en = Serial.read();
        if (en == 'e') {
            digitalWrite(led, 1);
        }
        else if (en == 'a')
            digitalWrite(led, 0);
    }
    digitalWrite(Trigger, HIGH);
    delayMicroseconds(10); //Enviamos un pulso de 10us
}
```

```

digitalWrite(Triple, LOW);
t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
d = t / 59;              //escalamos el tiempo a una distancia en cm

Serial.write(d);         //Enviamos al serial el valor de la distancia
delay(100);              //Hacemos una pausa de 100ms

}

```

Líneas del código Processing (Java).

```

import processing.serial.*;
Serial aport;
PImage photo;
PImage photo2;
int l = 0;
int v = 0;
int num = 20;
int puerto;
float pos = 0;
float mx[] = new float[num];
float my[] = new float[num];

void setup(){
  aport = new Serial(this, "COM3", 9600);
  size(900, 400);
  photo = loadImage("logouv.jpg");
  photo2 = loadImage("instru.png");
  noStroke();
  fill(255, 153);
}

void draw(){
  background(0);
  while (aport.available() > 0) {           // Consulta si hay un byte disponible
  en el puerto
    puerto = aport.read();
    pos=map(puerto,10,160,412,792);
  }
  photo.resize(70, 80);
  image(photo, 0, 0);
  photo2.resize(70, 80);
  image(photo2, 80, 0);
  fill(#FFFFFF);
  textSize(50);
  text("EQUIPO NUMERO 6",180,50);
  if((mouseX > 180) && (mouseX < 285) && (mouseY > 280) && (mouseY < 320)){
    fill(#20CB15);
  }
  if(mousePressed){
    fill(#0EE000);
    aport.write('e');
    l = 1;
  }
}

```

```

}
else{
    fill(#13830C);
}
rect(180, 280, 105, 40, 8);

fill(#FFFFFF);
textSize(20);
text("ON", 215, 350);

if((mouseX > 40) && (mouseX < 145) && (mouseY > 280) && (mouseY < 320)) {
    fill(#C91616);
if(mousePressed){
    fill(#DE2C00);
    aport.write('a');
    l=0;
}
}
else{
    fill(#810D0D);
}
rect(40, 280, 105, 40, 8);

fill(#FFFFFF);
textSize(20);
text("OFF", 70, 350);

if(l==1){
    fill(#FFFFFF);
    textSize(30);
    text("Estado del led: Encendido", 0, 200);
}
else{
    fill(#FFFFFF);
    textSize(30);
    text("Estado del led: Apagado", 0, 200);
}

if(puerto > 160){
    v=792;
}
else if(puerto < 10){
    v=412;
}
else {
    v=int(pos);
}
int which = frameCount % num;
mx[which] = v;
my[which] = 255;

for (int i = 0; i < num; i++) {
    // which+1 is the smallest (the oldest in the array)
    int index = (which+1 + i) % num;

```


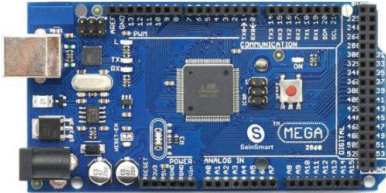
```



    ellipse(mx[index], my[index], i, i);
}
noFill();
strokeWeight(4);
stroke(#FFFFFF);
    rect(400, 240, 404, 30, 13 );
    if(puerto > 159){
        fill(#11E000);
        textSize(30);
        text("Objeto fuera de rango", 440, 200);
    }
    else{
        fill(#DEDBDB);
        textSize(30);
        text("Distancia al objeto "+puerto+"cm", 410, 200);
    }

    if(puerto < 10){
        fill(#FF0505);
        textSize(40);
        text(";OBJETO DEMASIADO CERCA!", 310, 350);
    }
}

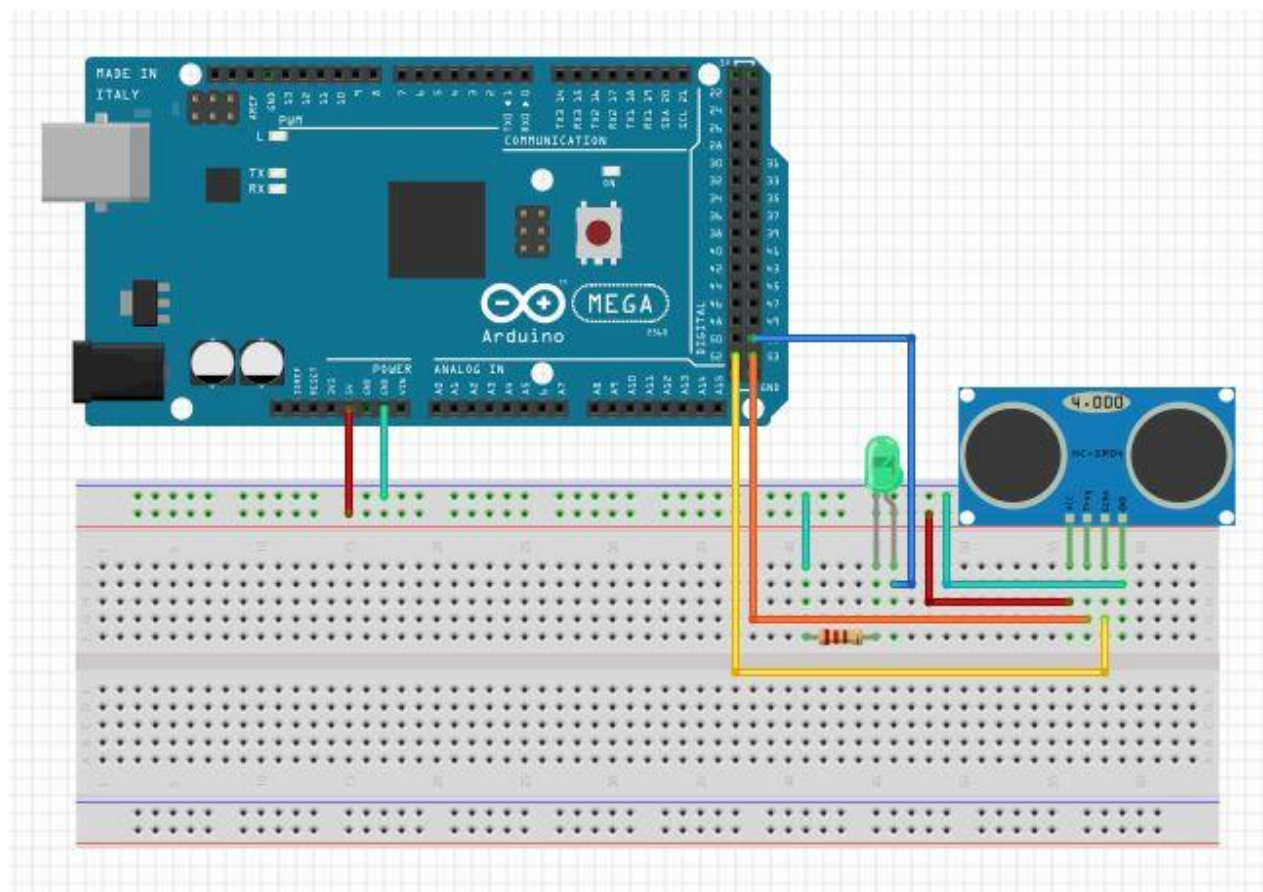
```

3. LISTA DE COMPONENTES

COMPONENTE	IMAGEN	DESCRIPCIÓN
Sensor ultrasónico hc-sr04		<p>Sensor el cual consta principalmente de 4 pines (Vcc, Trigger, Echo, Gnd)</p> <p>Y dos transductores uno para enviar pulsos ultrasónicos y el otro los recibe después de encontrarse con un objeto.</p>
Arduino Mega		<p>Placa de 8 bits con 54 pines digitales, 16 entradas analógicas y 4 puertos seriales.</p> <p>Utilizamos esta placa especialmente debido a que es capaz de realizar más de 2</p>

		interrupciones (en los pines 2,3,21,20,19,18)
Un LED		Fuente de luz constituida por un material semiconductor dotado de dos terminales.
Resistencias		Componente electrónico diseñado para introducir una resistencia eléctrica determinada entre dos puntos de un circuito eléctrico.

4. DIAGRAMA



5. CONCLUSIONES

Al llevar a cabo este proyecto pude notar como es que existen una infinidad de herramientas las cuales se pueden aplicar para desarrollar una UI cada una con sus pros contras limitaciones ventajas así como un lenguaje distinto al momento de programar también pude darme cuenta de que existen otras maneras de usar los distintos sistemas embebidos y distintos firmwares con los cuales se puede facilitar la programación y hacer que los proyectos desarrollados sean fácilmente replicados y se puedan adaptar a las distintas placas del mercado.

6. BIBLIOGRAFÍA

- [1] Reference. Processing [Online]. Disponible en: <https://processing.org/reference/>
- [2] MCI Capacitación- MCI Electronics (Jun 11, 2019). Conectando Arduino a Processing [Online]. Disponible en: <https://cursos.mcielectronics.cl/2019/06/11/conectando-arduino-a-processing/>