

ANEXO 7 ±PROGRAMACIÓN ARDUINO.

```
#include "ArduinoJson.h" //Generador
de Json para C++, el asistente se pue
de encontrar aca https://arduinojson.
org/v5/assistant/
//LED MULTICOLOR
int red = 6;
int blue = 3;
int green = 5;
int objbueno = 13;
int objmalo = 8;
int datoumbral = 0;
// INICIO Y PARO
int emerg = 9;
int start = 7;
int ledcamara = 4;

//Variables para iniciar proceso y pa
rar el proceso
bool inicioproceso = false;
bool paroemergencia = false;
bool modofuncionamiento = true;
int datosentidomotor = 0;
int datotoclasificar = 0;
int datoestado = 0;
bool stanby = false;
bool iluminacion = false;
//Variables LED rgb
int rgb0 = 255; //Rojo
int rgb1 = 255; //Verdel
int rgb2 = 255; //Azul
bool sis_activado = false;
bool sis_activado2 = false;
bool resetdatosentido = false;
int receptor = 0;
int umbral = 450;
#include <Servo.h>
Servo clasificador;
int pinservo = 10;
#define motorpwm 11
int motordir = 12;
bool actd = false;
bool antd = false;
bool vidprocesado = false;
bool clasificador servo = false;
int dato = 70;
int datoclasificacion = 80;
int valorinicial = 0; //Valor de la e
ntrada infrarrojo
int valor = 0; //Valor de la entrada
infrarrojo
int dispositivo = 0; //Identificador
dispositivo
char c, d;
bool centrado = false;
//Variables para Arduino JSON Generat
or/Parser
```

```
const size_t bufferSize = JSON Objec
T_SIZE(11);
DynamicJsonBuffer Buffersalida(buffer
Size);
JsonObject& datosalidajson = Buffersa
lida.createObject();
unsigned long tiempoinicio = 0;
String datospuertoserial = "";
bool haycadena = false;
bool iniciocadena = false;
bool fincadena = false;
int estado = 0;
bool iniciarmanual = false; //Inicia
una vez el modo manual
#define debug 1
void setup() {
  Serial.begin(115200);
  pinMode(red, OUTPUT);
  pinMode(blue, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(objmalo, OUTPUT);
  pinMode(objbueno, OUTPUT);
  pinMode(motorpwm, OUTPUT);
  pinMode(motordir, OUTPUT);
  pinMode(emerg, INPUT);
  pinMode(start, INPUT);
  pinMode(ledcamara, OUTPUT);
  digitalWrite(motordir, HIGH);
  clasificador.attach(pinservo);
  clasificador.write(110);
  analogWrite(motorpwm, 0);
  analogWrite(red, 255 - rgb0);
  analogWrite(blue, 255 - rgb1);
  analogWrite(green, 255 - rgb2);
  valor = analogRead(receptor);
  Serial.setTimeout(1000);
  /*solo para manual de momento*/
  if ( (valorinicial >= 0) || (valori
nicial <= 1000)) {
    sis_activado = false;
  } else {
    sis_activado = true;
  }
  if ( valor <= umbral) {
    actd = true;
  } else {
    actd = false;
  }
  datosalidajson["presencia"] = actd;
  if (digitalRead(emerg) == LOW) {
    paroemergencia = true;
  }
  else paroemergencia = false;
  if (paroemergencia == true) {
    estado = -1;
    datosalidajson["estado"] = -1;
```

```

    } else {
        estado = 0;
        datosalidajson["estado"] = 0;
    }
    antd = actd;
    datosalidajson.printTo(Serial);
}
void loop() {
    if (Serial.available() > 0) {
        // Serial.print("Datos serial:");
        Serial.print(Serial.available());
    }
    while (Serial.available() > 0) { //Si
        empre leo los datos del Serial
        haycadena = true;
        d = Serial.read();
        //Verifico que me lleguen las llaves { y } antes de procesar el objeto
        if (d == '{') {
            datospuertoserial = "";
            iniciocadena = true;
        }
        if (iniciocadena) {
            datospuertoserial.concat(d);
            rgb0 = 20, rgb1 = 20, rgb2 = 20;
            analogWrite(red, 255 - rgb0);
            analogWrite(blue, 255 - rgb1);
            analogWrite(green, 255 - rgb2);
        }
        if (d == '}') {
            fincadena = true;
            break;
        }
    } //Fin del Serial Available
    if (iniciocadena && fincadena) {
        iniciocadena = false, fincadena = false;
        rgb0 = 255, rgb1 = 255, rgb2 = 0;
        analogWrite(red, 255 - rgb0);
        analogWrite(blue, 255 - rgb1);
        analogWrite(green, 255 - rgb2);
        delay(200);
        DynamicJsonBuffer Bufferentrada(buff
erSize);
        JsonObject& datojson = Bufferentrada
.parseObject(datospuertoserial);
        if (datojson.success()) {
            rgb0 = 255, rgb1 = 255, rgb2 = 0;
            analogWrite(red, 255 - rgb0);
            analogWrite(blue, 255 - rgb1);
            analogWrite(green, 255 - rgb2);
            delay(200);
            for (auto kv : datojson) {
                if (String(kv.key).compareTo(
"dispositivo") == 0) {
                    dispositivo = kv.value.as<unsigned
int>();
                }
            }
        }
    }
}

```

```

        else if (String(kv.key).compareTo
("modofunc") == 0) {
            modofuncionamiento = kv.value.as<
bool>();
            if (modofuncionamiento) Serial.pr
int("auto");
        } else {
            iniciarmanual = true;
            Serial.println("manual");
        }
    }
    else if (String(kv.key).compareTo
("motor") == 0) {
        dato = kv.value.as<signed int>();
        datoclasificacion = dato;
    }
    else if (String(kv.key).compareTo
("estado") == 0) {
        datoestado = kv.value.as<signed
int>();
        if (datoestado == -1) {
            paroemergencia = true;
            inicioproceso = false;
            rgb0 = 120, rgb1 = 0, rgb2 = 0;
            analogWrite(motorpwm, 0);
            if (estado != -1) {
                estado = -1;
                datosalidajson["estado"] = -1;
                datosalidajson.printTo(Serial);
            }
        }
        else if (datoestado == 0) {
            paroemergencia = false;
            inicioproceso = false;
            centrado = false;
            rgb0 = 255, rgb1 = 69, rgb2 = 0;
            //Naranja
            analogWrite(motorpwm, dato);
            if (estado != 0) {
                estado = 0;
                datosalidajson["estado"] = 0;
                datosalidajson.printTo(Serial);
            }
        }
        else if (datoestado == 1) {
            if (estado != 1) {
                estado = 1;
                datosalidajson["estado"] = 1;
                datosalidajson.printTo(Serial);
            }
            centrado = false;
            paroemergencia = false;
            inicioproceso = true;
            dato = datoclasificacion;
            rgb0 = 0, rgb1 = 100, rgb2 = 0;
            //Verde
        }
    }
}

```

```

    else if (String(kv.key).compareTo(
("clasificador") == 0) {
    clasificador servo = kv.value.as<boolean>();
    }
    else if (String(kv.key).compareTo("
imagenprocesada") == 0) {
    rgb0 = 255, rgb1 = 255, rgb2 = 255;
    analogWrite(red, 255 - rgb0);
    analogWrite(blue, 255 - rgb1);
    analogWrite(green, 255 - rgb2);
    vidprocesado = kv.value.as<boolean>();
    Serial.print("vidprocesado:"); Serial.print(vidprocesado);
    }
    else if (String(kv.key).compareTo("sentidomotor") == 0 ) {
    datosentidomotor = kv.value.as<signed int>();
    }
    else if (String(kv.key).compareTo("direcclassificacion") == 0 ) {
    datotoclasificar = kv.value.as<signed int>();
    }
    else if (String(kv.key).compareTo("umbral") == 0) {
    datoumbral = kv.value.as<signed int>();
    }
    else if (String(kv.key).compareTo("capturarengre") == 0) {
    iluminacion = kv.value.as<boolean>();
    }
    } //Fin del For
    datospuertoserial = "";
    } //Fin del procesamiento de JSON
    else {
    Serial.print("Falla JSON"); Serial.println(datospuertoserial); //Si por algun motivo no procesa bien, devuelva al dispositivo que genero la señal
    }
    datospuertoserial = "";
    }
    //Fin de procesamiento de JSON
    if (digitalRead(emerg) == LOW) {
    centrado = false;
    rgb0 = 255, rgb1 = 0, rgb2 = 0; //ROJO
    analogWrite(red, 255 - rgb0);
    analogWrite(green, 255 - rgb1);
    analogWrite(blue, 255 - rgb2);

    inicioproceso = false;
    paroemergencia = true;

```

```

    }
    /*else if (paroemergencia == true &
    & digitalRead(emerg) == HIGH) {
    paroemergencia = true;
    }*/
    else {
    //Serial.print("emergenciaoff");
    paroemergencia = false;
    }
    if (digitalRead(start) == HIGH) {
    delay(500);
    if ( digitalRead(start) == HIGH)
    {
    rgb0 = 255, rgb1 = 69, rgb2 = 0
    ; //Naranja
    inicioproceso = false;
    centrado = false;
    datosalidajson["estado"] = 0;
    delay(1000);
    }
    else {
    delay(500);
    inicioproceso = true;
    dato = 80;
    datosalidajson["estado"] = 1;
    rgb0 = 0, rgb1 = 100, rgb2 = 0;
    }
    }

    if (paroemergencia) {
    //Serial.print("emergencia");
    analogWrite(motorpwm, 0);
    inicioproceso = false;
    if (estado != -1) {
    estado = -1;
    datosalidajson["estado"] = -1;
    datosalidajson.printTo(Serial);
    }
    //
    aqui añado
    analogWrite(red, 255 - rgb0);
    analogWrite(green, 255 - rgb1);
    analogWrite(blue, 255 - rgb2);
    // paroemergencia = false; //aqui añado
    }
    else { //Else Emergencia
    ///Serial.print("elseparoemergencia");
    if (inicioproceso && estado != 1
    && !paroemergencia ) {
    Serial.print("run");
    estado = 1;
    datosalidajson["estado"] = 1;
    datosalidajson.printTo(Serial);
    }
    else if (!inicioproceso && estado
    != 0 && !paroemergencia ) {
    Serial.print("stop");

```

```

    rgb0 = 255, rgb1 = 69, rgb2 = 0
; //Naranja
    estado = 0;
    datosalidajson["estado"] = 0;
    datosalidajson.printTo(Serial);

}
if (digitalRead(start) == HIGH) {
    delay(500);
    if ( digitalRead(start) == HIGH
) {
        rgb0 = 255, rgb1 = 69, rgb2 =
0; //Naranja
        inicioproceso = false;
        centrado = false;
        datosalidajson["estado"] = 0;
        delay(1000);
    }
    else {
        delay(500);
        inicioproceso = true;
        dato = 80;
        datosalidajson["estado"] = 1;
        rgb0 = 0, rgb1 = 100, rgb2 = 0;
    }
}
}
if (modofuncionamiento == true) {
    //Serial.print("a");
    if (inicioproceso) {
        Serial.print("i");
        digitalWrite(ledcamara, LOW);
        analogWrite(red, 255 - rgb0);
        analogWrite(green, 255 - rgb1);
        analogWrite(blue, 255 - rgb2);
        analogWrite(motorpwm, dato);
        valor = analogRead(receptor);
        if ( valor <= umbral) {
            actd = true;
        }
        else {
            actd = false;
        }
        delay(300);
        if ( actd != antd) {
            datosalidajson["presencia"] = actd;
            if (inicioproceso) {
                datosalidajson["estado"] = 1;
            }
            else datosalidajson["estado"] = 0;
            datosalidajson.printTo(Serial);
            delay(100);
        }
        if (actd == true && vidprocesado =
= false) {
            digitalWrite(ledcamara, HIGH);
            dato = 0;
            analogWrite(motorpwm, dato);
            if (!centrado) {

```

```

                Serial.print("c");
                centrado = true;
                c = '0';
                tiempoinicio = millis();

                while ( c != '9' ) {
                    if (millis() - tiempoinicio > 100
00) break;
                    if (Serial.available()) {
                        c = Serial.read();
                        if (c == 'a') {
                            digitalWrite(motordir, HIGH);
                            analogWrite(motorpwm, 80);
                            rgb0 = 0, rgb1 = 255, rgb2 = 255;
                            analogWrite(red, 255 - rgb0);
                            analogWrite(green, 255 - rgb1);
                            analogWrite(blue, 255 - rgb2);
                            delay(100);
                            Serial.print('1');
                        }
                        else if (c == 'd') {
                            rgb0 = 125, rgb1 = 125, rgb2 = 125;
                            analogWrite(red, 255 - rgb0);
                            analogWrite(green, 255 - rgb1);
                            analogWrite(blue, 255 - rgb2);
                            digitalWrite(motordir, LOW);
                            analogWrite(motorpwm, 80);
                            delay(100);
                            Serial.print('1');
                        }
                        else if (c == '2') break;
                            analogWrite(motorpwm, 0);
                            digitalWrite(motordir, HIGH);
                            delay(5);
                        }
                    if (millis() - tiempoinicio > 10000)
break;
                        rgb0 = 0, rgb1 = 100, rgb2 = 100;
                        analogWrite(red, 255 - rgb0);
                        analogWrite(green, 255 - rgb1);
                        analogWrite(blue, 255 - rgb2);
                        if (digitalRead(emerg) == LOW) {
                            centrado = false;
                            paroemergencia = true;
                            if (estado != -1) {
                                break;
                                estado = -1;
                                datosalidajson["estado"] = -1;
                                datosalidajson.printTo(Serial);
                            } break;
                        }
                    if (digitalRead(start) == HIGH) {
                        delay(500);
                        if ( digitalRead(start)
== HIGH) {
                            Serial.println("R6vt");
                            rgb0 = 255, rgb1 = 69, rgb2 = 0;
//Naranja
                            inicioproceso = false;

```

```

centrado = false;
datosalidajson["estado"] = 0;
delay(1000);
break;
}
else {
delay(500);
modofuncionamiento = true;
inicioproceso = true;
dato = 80;
datosalidajson["estado"] = 1;
rgb0 = 0, rgb1 = 100, rgb2 = 0;
}
break;
}
}
Serial.print('2');
rgb0 = 0, rgb1 = 255, rgb2 = 0;
analogWrite(red, 255 - rgb0);
analogWrite(green, 255 - rgb1);
analogWrite(blue, 255 - rgb2);
c = 'f';
}
}
else if (vidprocesado == true) {
Serial.println("imagenprocesa
do");
vidprocesado = false;
centrado = false;
rgb0 = 255, rgb1 = 125, rgb2
= 125;
analogWrite(red, 255 - rgb0);
analogWrite(green, 255 - rgb1
);
analogWrite(blue, 255 - rgb2)
;
dato = datoclasificacion;
if (clasificadorservo) {
clasificador.write(110);
digitalWrite(objbueno, HIGH
);
digitalWrite(objmalo, LOW);
}
else {
clasificador.write(30);
digitalWrite(objbueno, LOW)
;
digitalWrite(objmalo, HIGH)
;
}
if (actd == true) {
valor = analogRead(receptor
);
if ( valor <= umbral) {
actd = true;
}
else {
actd = false;

```

```

}
digitalWrite(ledcamara, LOW);
analogWrite(motorpwm, dato);
digitalWrite(motordir, HIGH);
delay(1000);
}
rgb0 = 0, rgb1 = 255, rgb2 = 0;
}
antd = actd;
} else {
digitalWrite(ledcamara, LOW);
analogWrite(motorpwm, 0);
analogWrite(red, 255 - rgb0);
analogWrite(green, 255 - rgb1);
analogWrite(blue, 255 - rgb2);
}
}
else { // Aqui Hago el modo manual

}

}

digitalWrite(motordir, LOW);
delay(1000);
analogWrite(motorpwm, 0);
} else analogWrite(motorpwm, 0);
if (iluminacion == true) {
rgb0 = 20, rgb1 = 55, rgb2 = 0;
iluminacion = false;
Serial.println("camaraencendida: ");
digitalWrite(ledcamara, HIGH);
delay(2000);
digitalWrite(ledcamara, LOW);
} else digitalWrite(ledcamara,
LOW);

```