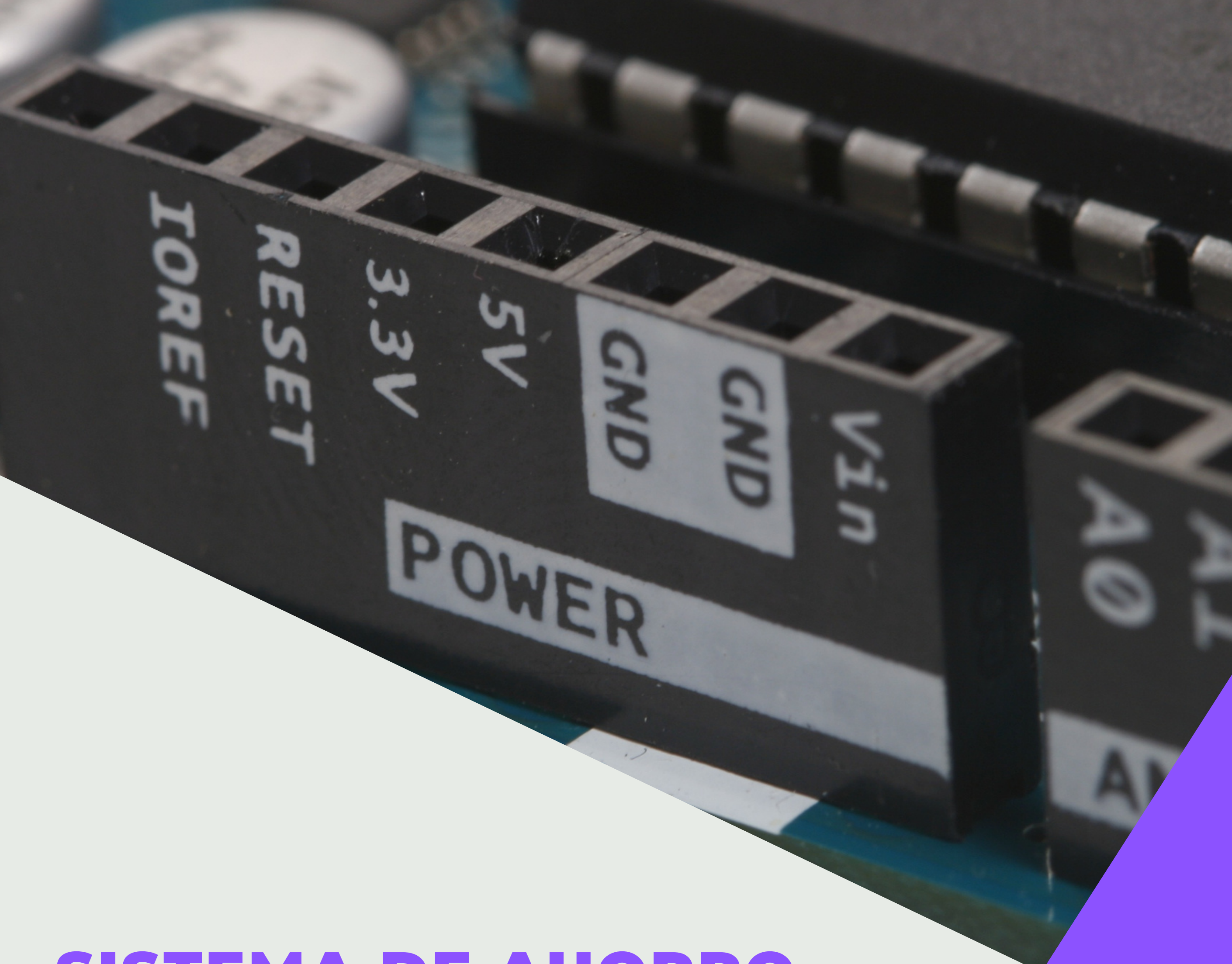


# UTEZ UNIVERSIDAD TECNOLÓGICA EMILIANO ZAPATA



## SISTEMA DE AHORRO DE ENERGÍA Y ANTIRROBO PRINCIPIOS DE IOT 4A

- DOMINGUEZ CASTAÑEDA ALAN  
MATTHEW ESTEBAN
- BAHENA LUIS EDUARDO
- HERRERA HERNANDEZ JOEL  
ALEJANDRO

Cuernavaca Morelos, 15/12/2022



## Tabla de contenido

Objetivo .....	2
Desarrollo .....	2
Código implementado: .....	1
Configuración del bluetooth HC-06 .....	1
Configuración del bluetooth HC-05 .....	1
Código de la alarma y ahorro de energía.....	4
Análisis del resultado .....	6
Conclusión.....	6

## Objetivo

Diseñar un sistema que permita ahorrar energía eléctrica y mandar una señal al usuario donde el espacio esté ocupado.

El ahorro de energía eléctrica debe ser programado:

A partir de que cierta hora del día, se encienda la luz.

Si el usuario se encuentra dentro de su hogar, y no está en movimiento, se debe apagar la luz.

Por otra parte, en cuanto la alarma antirrobo debe indicar:

Si el usuario no se encuentra en su domicilio, mandar un mensaje si detecta presencia y mandar a sonar una alarma. Mediante conexión bluetooth o wifi se debe notificar al usuario, esto se visualizará en el monitor serial, de tal manera que se hará uso de herramientas como dos arduinos, dos módulos bluetooth, una PC, un buzzer, dos LED's, resistencias, jumpers y protoboards.

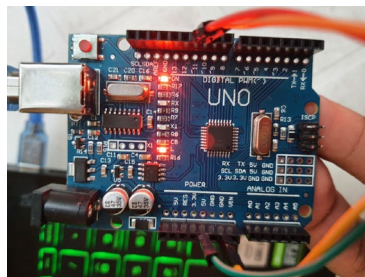
## Desarrollo

En primer lugar debemos lograr establecer una conexión entre los bluetooth. Se hicieron uso de dos módulos de diferente tipo, dado que uno emite la señal y el otro la recibe.



Primero se hizo uso de cables macho-hembra, asimismo se identificó los 4 pines que son:

1. **VCC:** Voltaje positivo de alimentación, soportando 5V
2. **GND,** Voltaje negativo de alimentación, se tienen que conectar al GND del Arduino.
3. **TX:** Pin de Transmisión de datos, por este pin el HC-06 transmite los datos que le llegan , conectado al puerto 11 del Arduino
4. **RX:** Pin de Recepción, conectado al puerto 10 del Arduino



*III-Ubicación de los cables*

Código implementado:

```
#include <SoftwareSerial.h>

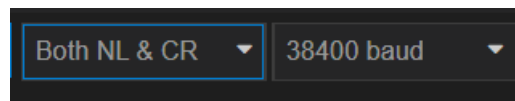
SoftwareSerial BT(10, 11); //TXD (10), RXD(11)

void setup() {
  Serial.begin(38400);
  BT.begin(38400);
  Serial.println("Conexion lista");
}

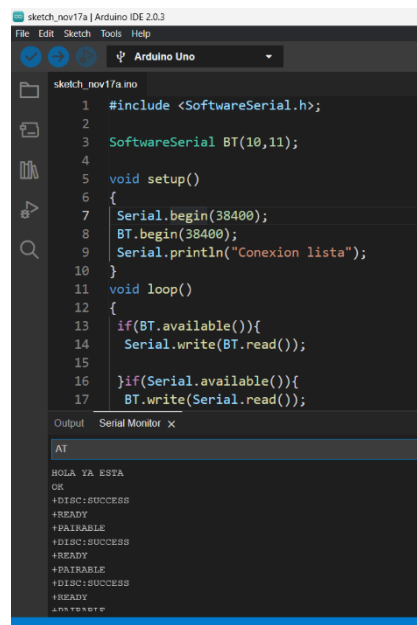
void loop() {
  if (BT.available()) {
    Serial.write(BT.read());
  }
  if (Serial.available()) {
    BT.write(Serial.read());
  }
}
```

#### IV-Código Arduino

**Nota:** El Serial debe estar en 38400 y Both NL & CR



Configuración del bluetooth HC-06



Comunicación comando AT: Para saber si hay comunicación con el modulo

```

sketch_nov17a | Arduino IDE 2.0.3
File Edit Sketch Tools Help
sketch_nov17a.ino
1 #include <SoftwareSerial.h>;
2
3 SoftwareSerial BT(10,11);
4
5 void setup()
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

```

AT+NAME
-----
+DISC:SUCCESS
+READY
+PAIRABLE
+DISC:SUCCESS
+READY
+PAIRABLE
+DISC:SUCCESS
+READY
+PAIRABLE
+NAME=ALEX
  
```

```

sketch_nov17a | Arduino IDE 2.0.3
File Edit Sketch Tools Help
sketch_nov17a.ino
1 #include <SoftwareSerial.h>;
2
3 SoftwareSerial BT(10,11);
4
5 void setup()
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

```

AT+NAMEPACAJONES
-----
+NAME=ALEX
+DISC:SUCCESS
+READY
+PAIRABLE
+NAME=PACAJONES
OK
+DISC:SUCCESS
+READY
+PAIRABLE
+DISC:SUCCESS
  
```

Nombre AT+NAME: Mostrar nombre

AT+NAMEPACAJONES: Asignar nombre

```

sketch_nov17a | Arduino IDE 2.0.3
File Edit Sketch Tools Help
sketch_nov17a.ino
1 #include <SoftwareSerial.h>;
2
3 SoftwareSerial BT(10,11);
4
5 void setup()
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

```

AT+BAUD
-----
+NAME=PACAJONES
OK
+DISC:SUCCESS
+READY
+PAIRABLE
+DISC:SUCCESS
+READY
+PAIRABLE
+NAME=PACAJONES
+BAUD=6
  
```

Velocidad de la comunicación AT+BAUD<numero>

Número---baudrate

1	-----1200	5	-----19200
2	-----2400	6	-----38400
3	-----4800	7	-----57600
4	-----9600	8	-----115200

```

sketch_nov17a | Arduino IDE 2.0.3
File Edit Sketch Tools Help
Arduino Uno
sketch_nov17a.ino
1 #include <SoftwareSerial.h>;
2
3 SoftwareSerial BT(10,11);
4
5 void setup()
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

AT+PIN

+READY  
+PAIRABLE  
+DISC+SUCCESS  
+READY  
+PAIRABLE  
+NAME=PACCOJONES  
+BAUD=6  
+BAUD=6  
+OK  
+PIN=4321

```

sketch_nov17a | Arduino IDE 2.0.3
File Edit Sketch Tools Help
Arduino Uno
sketch_nov17a.ino
1 #include <SoftwareSerial.h>;
2
3 SoftwareSerial BT(10,11);
4
5 void setup()
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

AT+PIN0987

+DISC+SUCCESS  
+READY  
+PAIRABLE  
+NAME=PACCOJONES  
+BAUD=6  
+BAUD=6  
+OK  
+PIN=4321  
+PIN=0987  
+OK

## Cambiar Código de vinculación AT+PIN

```

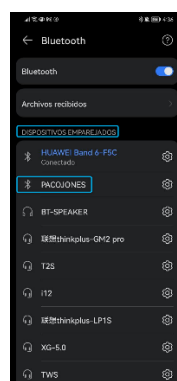
sketch_nov17a | Arduino IDE 2.0.3
File Edit Sketch Tools Help
Arduino Uno
sketch_nov17a.ino
1 #include <SoftwareSerial.h>;
2
3 SoftwareSerial BT(10,11);
4
5 void setup()
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

AT+VERSION

+PAIRABLE  
+NAME=PACCOJONES  
+BAUD=6  
+BAUD=6  
+OK  
+PIN=4321  
+PIN=0987  
+OK  
+PIN=0987  
+VERSION=V2.0.2

## Consultar versión del módulo AT+VERSION



## Configuración del bluetooth HC-05

```

sketch_nov17a.ino
1  #include <SoftwareSerial.h>;
2
3  SoftwareSerial BT(10,11);
4
5  void setup()
6  {
7    Serial.begin(38400);
8    BT.begin(38400);
9    Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

AT

HOLA YA ESTA  
OK

## Test de comunicació con AT

```

sketch_nov17a.ino
1  #include <SoftwareSerial.h>;
2
3  SoftwareSerial BT(10,11);
4
5  void setup()
6  {
7    Serial.begin(38400);
8    BT.begin(38400);
9    Serial.println("Conexion lista");
10 }
11 void loop()
12 {
  
```

Output Serial Monitor x

AT+NAME?

HOLA YA ESTA  
+NAME:ROOT  
OK

```

sketch_nov17a.ino
6  {
7    Serial.begin(38400);
8    BT.begin(38400);
9    Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

AT+NAME=Luis

HOLA YA ESTA  
+NAME:ROOT  
OK  
+NAME:Luis  
OK

## Cambiar nombre de nuestro módulo AT+NAME? | AT+NAME=<name>

```

sketch_nov17a.ino
6  {
7    Serial.begin(38400);
8    BT.begin(38400);
9    Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

AT+PSWD?

HOLA YA ESTA  
+NAME:ROOT  
OK  
+NAME:Luis  
OK  
+PIN:"1234"  
OK

```

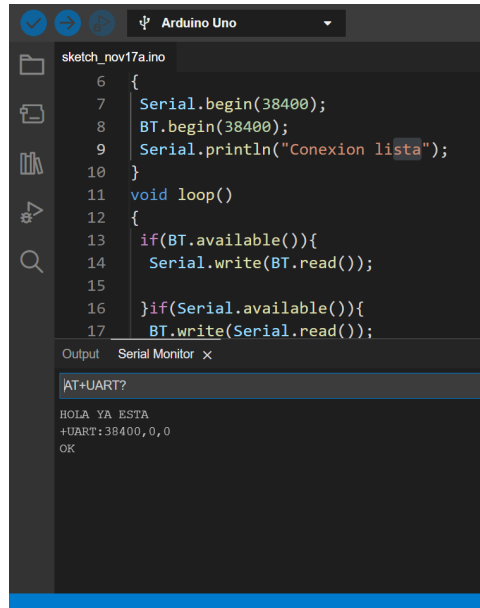
sketch_nov17a.ino
6  {
7    Serial.begin(38400);
8    BT.begin(38400);
9    Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
  
```

Output Serial Monitor x

AT+PSWD="2003"

HOLA YA ESTA  
+PIN:"2003"  
OK

## Cambiar código de vinculación AT+PSWD? | AT+PSWD=""



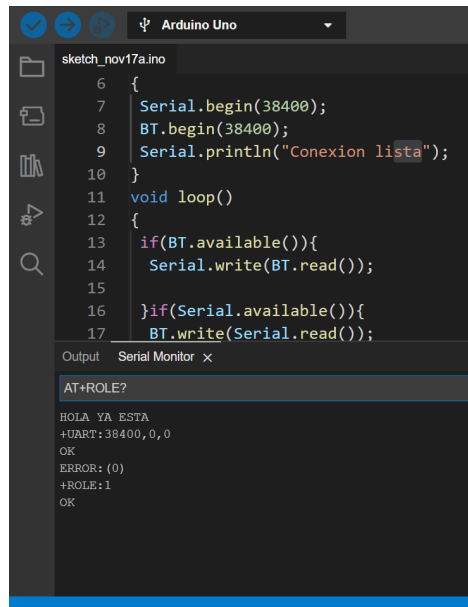
```
sketch_nov17a.ino
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
18   }
19 }
```

Output Serial Monitor x

AT+UART?

HOLA YA ESTA  
+UART:38400,0,0  
OK

Velocidad de comunicación: AT+UART?



```
sketch_nov17a.ino
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
18   }
19 }
```

Output Serial Monitor x

AT+ROLE?

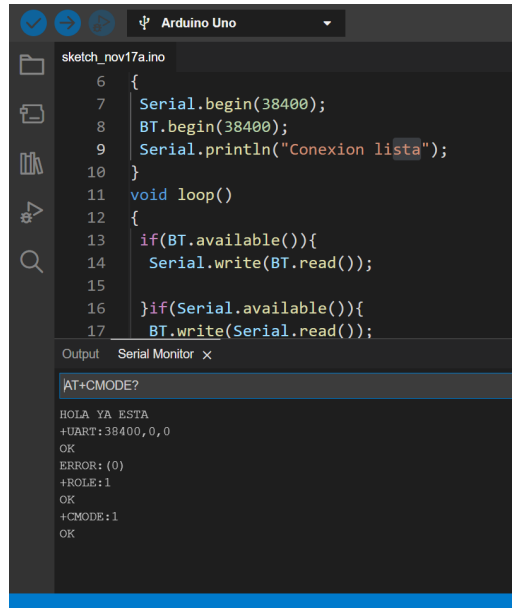
HOLA YA ESTA  
+UART:38400,0,0  
OK  
ERROR: (0)  
+ROLE:1  
OK

Role del modulo: AT+ROLE?

0 -> Esclavo

1 -> Maestro





```
sketch_nov17a.ino
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
18   }
19 }
```

Output Serial Monitor x

AT+CMODE?

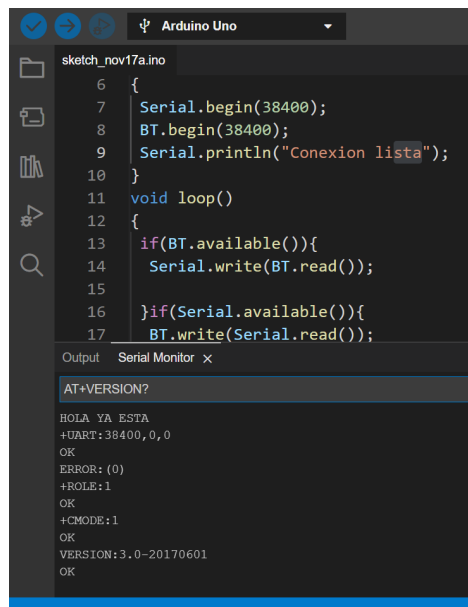
HOLA YA ESTA  
+UART:38400,0,0  
OK  
ERROR: (0)  
+ROLE:1  
OK  
+CMODE:1  
OK

Modo de conexión AT+CMODE?

0 -> Conectarse a un dispositivo con la dirección especificada(Se utiliza otro comando AT para especificar esta dirección).

1 -> conectar el módulo a cualquier dirección disponible(aleatorio).

AT+BIND= 98D3,41,F5D386 //Código del esclavo



```
sketch_nov17a.ino
6 {
7   Serial.begin(38400);
8   BT.begin(38400);
9   Serial.println("Conexion lista");
10 }
11 void loop()
12 {
13   if(BT.available()){
14     Serial.write(BT.read());
15   }
16   if(Serial.available()){
17     BT.write(Serial.read());
18   }
19 }
```

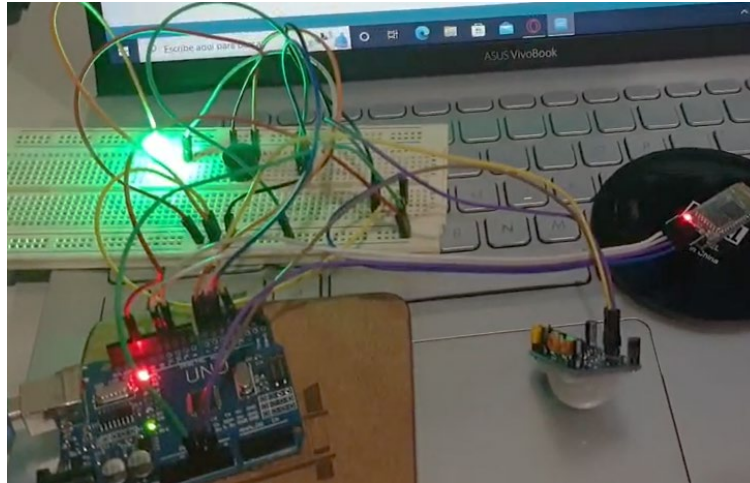
Output Serial Monitor x

AT+VERSION?

HOLA YA ESTA  
+UART:38400,0,0  
OK  
ERROR: (0)  
+ROLE:1  
OK  
+CMODE:1  
OK  
VERSION:3.0-20170601  
OK

Versión del modulo: AT-VERSION?

Posterior a esa configuración añadiremos en una protoboard con el bluetooth esclavo los demás elementos que son un buzzer, un sensor de obstáculo, y dos luces led, todo esto implementado de tal manera que están siendo alimentados y funcionan.



## Código de la alarma y ahorro de energía

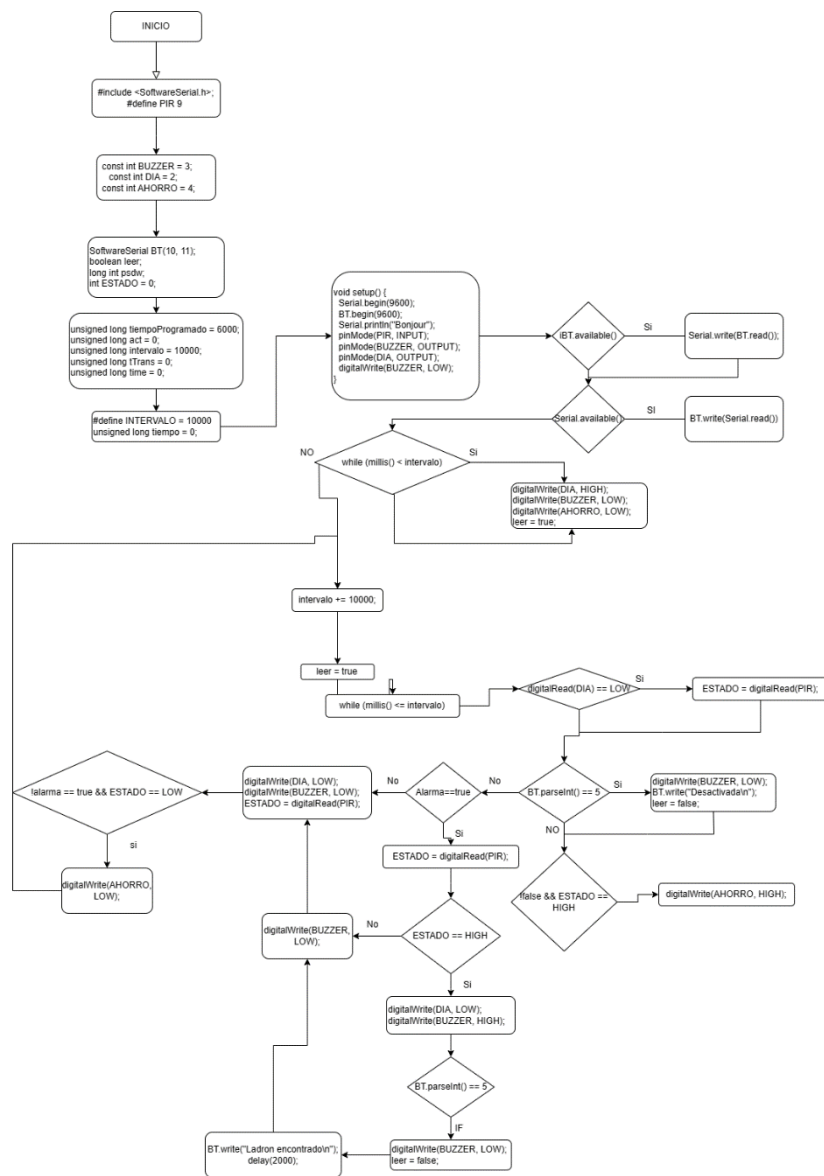
Por último, se tendrá la lógica que se cargará a él Arduino para que funcione.

```

BluetoothArno
1 #include <SoftwareSerial.h>
2 SoftwareSerial BT(10, 11); //TXD (10), RXD(11)  Amarillo rdx 11
3 #define PIR 9 //Entrada de la señal del sensor
4 const int BUZZER = 3; //salida para que suene el buzzer
5 const int DIA = 2; //luz led que representara el dia
6 const int AHORRO = 4; //Salida de luz led que sera el cuento de la casa
7 boolean alarma; //bandera para saber si hay contraseña o no
8 long int psdw;
9 const int ESTADO = 0; // almacena estado de señalint ESTADO = 0;
10 unsigned long tiempoProgramado = 6000;
11 unsigned long act = 0;
12 unsigned long intervalo = 10000; //el tiempo en el que comienza la alarma
13 unsigned long tTrans = 0;
14 unsigned long time = 0;
15 unsigned long tiempo = 0;
16
17
18 void setup() {
19   //Armo los seriales
20   Serial.begin(9600);
21   BT.begin(9600);
22   Serial.println("Bonjour");
23   pinMode(PIR, INPUT); // pin 3 como entrada del sensor
24   pinMode(BUZZER, OUTPUT);
25   pinMode(DIA, OUTPUT);
26   digitalWrite(BUZZER, LOW);
27   // Bind del bluetooth 9803,41,F5D386
28 }
29
30 void loop() {
31   if (BT.available()) {
32     Serial.write(BT.read()); //enviar de msj
33   }
34   if (Serial.available()) {
35     BT.write(Serial.read()); // alarm msj
36   }
37   //dia
38   while (millis() < intervalo) {
39     digitalWrite(DIA, HIGH); //solo prende la luz que representa el dia
40     digitalWrite(BUZZER, LOW);
41     digitalWrite(AHORRO, LOW);
42     alarma = true; //significa que no hay contraseña
43   }
44   intervalo += 10000;
45   digitalWrite(DIA, LOW); //apagamos la luz del dia
46   //noche
47   alarma = true;
48   while (millis() <= intervalo) {
49     if (digitalRead(DIA) == LOW) { //verificamos
50       ESTADO = digitalRead(PIR); //leemos el sensor
51       if (BT.parseInt() == 4) { //nuestra clave sera 4 para desactivar la alarma se lee por el serial del bluetooth
52         digitalWrite(BUZZER, LOW); //APAGAMOS
53         alarma = false;
54         BT.write("Desactivada\n"); // notificamos
55       } else if (!false && ESTADO == HIGH) { // Con contraseña y movimiento
56         digitalWrite(AHORRO, HIGH);
57       } else {
58       }
59     }
60   }
61 }
  
```

```

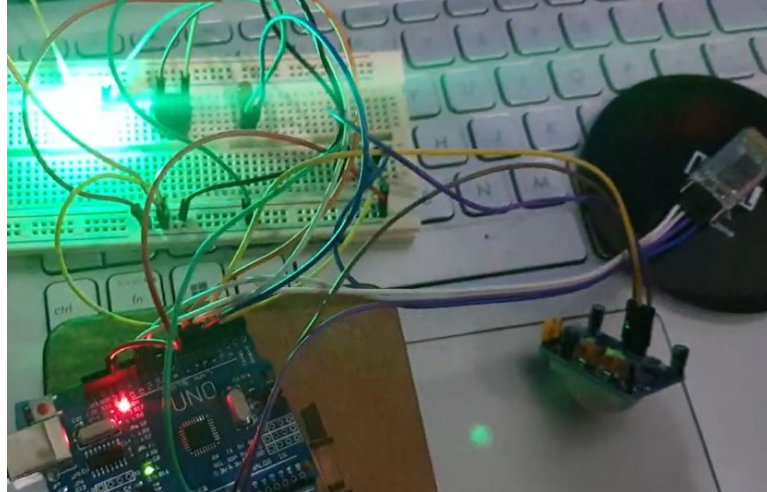
BluetoothArno
31 void loop() {
32   if (BT.available()) {
33     Serial.write(BT.read()); //enviar de msj
34   }
35   if (Serial.available()) {
36     BT.write(Serial.read()); // alarm msj
37   }
38   //dia
39   while (millis() < intervalo) {
40     digitalWrite(DIA, HIGH); //solo prende la luz que representa el dia
41     digitalWrite(BUZZER, LOW);
42     digitalWrite(AHORRO, LOW);
43     alarma = true; //significa que no hay contraseña
44   }
45   intervalo += 10000;
46   digitalWrite(DIA, LOW); //apagamos la luz del dia
47   //noche
48   alarma = true;
49   while (millis() <= intervalo) {
50     if (digitalRead(DIA) == LOW) { //verificamos
51       ESTADO = digitalRead(PIR); //leemos el sensor
52       if (BT.parseInt() == 4) { //nuestra clave sera 4 para desactivar la alarma se lee por el serial del bluetooth
53         digitalWrite(BUZZER, LOW); //APAGAMOS
54         alarma = false;
55         BT.write("Desactivada\n"); // notificamos
56       } else if (!false && ESTADO == HIGH) { // Con contraseña y movimiento
57         digitalWrite(AHORRO, HIGH);
58       } else {
59       }
60     }
61   }
62 }
  
```



## Análisis del resultado

Se muestra la funcionalidad del sistema en el siguiente enlace

<https://drive.google.com/file/d/13NCg53O4SmtbPkivm15BDKQPZpINyRXn/view?usp=sharing>



*V Imagen demostrativa*

## Conclusión

Para la realización de este proyecto se enfrentaron diversos retos, el principal fue la lógica del programa, algunos materiales desconocidos y a su curva de aprendizaje de los mismos, así como recalcar el corto periodo de tiempo para su realización.

Para facilitar el desarrollo de la lógica se programan por bloques de códigos, probándolos de manera independiente, así cuando cumplían con su acción correctamente se implementaba en el proyecto grande.

En pocas palabras, se logro desarrollar la alarma de manera correcta, solo modulando las acciones de lo pedido y probando diversas manera de conectar los componentes.