

ANEXO N° 1. CÓDIGO PARA CONFIGURACIÓN DE MÓDULO BLUETOOTH POR MEDIO DE MICROCONTROLADOR ATMEGA328P

```
#include <SoftwareSerial.h>
#define Tx 10
#define Rx 11
SoftwareSerial Bluetooth(Rx, Tx); // Recepción | Transmisión

void setup()
{
  pinMode(9, OUTPUT); // Este pin se conectará al pin 34 del módulo HC-05 para entrar a
  la opción de configuración por comandos AT
  digitalWrite(9, HIGH);
  Serial.begin(9600);
  Serial.println("Ingrese comandos AT:");
  Bluetooth.begin(9600);
}

void loop()
{
  // Si llega un dato por el puerto BT se envía al monitor serial
  if (Bluetooth.available())
    Serial.write(Bluetooth.read());

  // Si llega un dato por el monitor serial se envía al puerto BT
  if (Serial.available())
    Bluetooth.write(Serial.read());
}
```

ANEXO N° 2. PROGRAMACIÓN DEL MICROCONTROLADOR ATMEGA328P

```
#include <SoftwareSerial.h> // Se utilizará comunicación Serial con el Módulo Bluetooth

#define TxD 7 // Pines de comunicación que se utilizará para la comunicación
#define RxD 6 // 7 Tx al BT, 6 Rxd del BT

#define ALARMA 12 // Pin de Alerta de Desconexión de Electrodo

#define loffp 11 // Pines digitales de Electroodos

#define loffm 10

#define qrs 2 //Entrada de la señal cuadrada, con este pin se generarán las
interrupciones

#define botonpanico 3 // segunda interrupcion al presionar boton de panico

byte condicion = 0;

byte umbralFA = 70; // Valor umbral configurado con el valor de 70

byte ecg[1000]; // Matriz donde se almacenarán permanentemente 1000 muestras con
10 segundos de ECG

int i= 0;
int j = 0;
byte k = 0;
int m = 0;
byte n = 0;
byte p = 0;

//Variables para la implementación del algoritmo de decisión:
unsigned long t[10] = {0,0,0,0,0,0,0,0,0,0};
float RR[10] = {0,0,0,0,0,0,0,0,0,0};
float RRconrayita[10] = {0,0,0,0,0,0,0,0,0,0};
float RRnorm[10] = {0,0,0,0,0,0,0,0,0,0};
float desviacion[10] = {0,0,0,0,0,0,0,0,0,0};
float suma = 0;
float sumadesviacion = 0;
```

```
float varianza = 0;

float promedio = 0;

//

byte calcular =0;
byte panicodetectado = 0;
long T0 = 0;

// Inicio de Comunicación Serial hacia Módulo Bluetooth
SoftwareSerial Bluetooth(TxD,RxD);

void setup() {

    Bluetooth.begin(9600); //Velocidad configurada a 9600 baudios

//Configuración de pines de entrada y salida
    pinMode(loffp, INPUT);
    pinMode(loffm, INPUT);
    pinMode(qrs, INPUT);
    pinMode(botonpanico, INPUT);
    pinMode(ALARMA, OUTPUT);

    Serial.begin(9600); //Comunicación serial con PC para visualización de datos

//Configuración de interrupciones
// 1era: Con cada flanco de subida, esto es, con cada QRS detectado
// 2da: Al presionarse el botón de pánico
    attachInterrupt(digitalPinToInterrupt(qrs), Mostrar, RISING);
```

```
attachInterrupt(digitalPinToInterrupt(botonpanico), Panico, FALLING);

}

void loop() {

  if ((digitalRead(loffp) == 1) || (digitalRead(loffm) == 1)) { //Me fijo si los electrodos
estén conectados

    digitalWrite(ALARMA, HIGH); //Enciendo el LED de electrodo desconectado

    delay(100);

    Bluetooth.flush();

    Bluetooth.write(127); //127 Envío código de desconexion de electrodos

    do {delay(1000);} //No hago ninguna medición hasta que se vuelva a conectar el(los)
electrodo(s) desconectado(s)

    while ((digitalRead(loffp) == 1) || (digitalRead(loffm) == 1));

  }

  else { // Si electrodos están conectados, ejecuto el programa normal

    digitalWrite(ALARMA, LOW); // Apago la alarma

    if (i<1000) //Guardo 10s de información, 1000 muestras cada 10ms
    {
      ecg[i] = analogRead(A0)/3;
      delay(10); // Tomo una muestra cada 10ms
      i++;
    }

    //Si no se presiona el botón de pánico, se actualiza el conjunto de muestras cada 10ms
```

```
else if (panicodetectado == 0){
    for (m=0; m<=999; m++) {ecg [m] = ecg[m+1];
    ecg[999] = analogRead(A0); //
    delay(10);
    }
}

//Si se presionó el botón de pánico, se genera la imagen y se envía por BT
else if (panicodetectado == 1){

    generoimagen:  if (j<=999) //Se envían las 1000 muestras mapeadas por BT
    {
        ecg[j] = map(ecg[j], 50,250,100,1);
        Bluetooth.begin(9600);
        Bluetooth.write(ecg[j]);
        Bluetooth.end();
        j++;
    }

    else if (j >= 1000)
        { Bluetooth.write(126); //127 Código de evento generado o botón de alerta
        presionado
        j=0;
        i=0;
        panicodetectado=0;
        }

    else {delay(100);}
}
```

```
    }  
    }  
  
}  
  
//Interrupción por cada QRS detectado  
void Mostrar() {  
  
    //Serial.println("Pulso");  
    t[k] = millis();  
  
    if (k>0)  
    { RR[k] = millis() - t[k-1]; //tiempo entre cada interrupción = intervalo RR  
  
    // Luego de 10 complejos QRS detectado, se realizará la normalización de intervalos  
    RR  
    if (RRconrayita[9] == 0){RRconrayita[0] = RR[1];}  
    RRconrayita[k] = 0.75*RRconrayita[k-1]+0.25*RR[k];  
    RRnorm[k] = (RR[k])/(RRconrayita[k]); //Se debe eliminar, creo  
    RRnorm[k]= RR[k]*100;  
    RRnorm[k] = (RRnorm[k])/(RRconrayita[k]);  
    }  
    Serial.println(k);  
    k++;  
  
    // Asimismo, se calculará las varianzas de los valores normalizados, que será el  
    decisor directo  
    if (k==10)  
    {k=1;  
    t[0]=t[9];
```

```
RRconrayita[0] = RRconrayita[9];
for (byte n=1; n<=9; n++)
{suma = RRnorm[n]+suma;}
promedio = suma/9;
//Serial.println("El promedio es: ");
//Serial.println(promedio);
suma = 0;
for (byte k=1; k<=9; k++)
{desviacion[k] = pow(abs(RRnorm[k]-promedio),2);
sumadesviacion = desviacion[k]+sumadesviacion;
//Serial.println("Las desviaciones son: ");
//Serial.println(desviacion[k]);
}
varianza = sumadesviacion/9;
sumadesviacion = 0;
//Serial.print("La varianza es: ");
Serial.println(varianza);
//Serial.print("La frecuencia es ");
Serial.println((480000)/(t[9]-t[1]));

//Si se supera el valor umbral, se mandará a generar la imagen con 1000 muestras
if (varianza > umbralFA & condicion == 0) {condicion = 1; panicodetectado = 1;}

}

}

// Interrupción de botón de pánico presionado
void Panico() {
```

```
if (millis())>T0+250)
{
    T0 = millis();
    panicodetectado =1;

    //Serial.println("Boton de panico presionado");
}

}
```



Pantalla 1:

```
when Button1 .Click
do
  initialize local Variables to ""
  in
    set Variables to join
      TextBox1 . Text
      ""
      TextBox2 . Text
    open another screen with start value screenName "Screen2"
      startValue get Variables
```



Pantalla 2:

```

when Screen2.Initialize
do
  set global ejeX to 1
  set global ejeprevio to 1
  set global ejeY to 1
  set global dato to 1
  set global fila to 0
  set global medico to select list item list index 1
  set global cuidador to select list item list index 2
  call ActivityStarter1.StartActivity

when ListPicker1.BeforePicking
do
  set ListPicker1.Elements to BluetoothClient1.AddressesAndNames

when Button1.Click
do
  if call BluetoothClient1.Connect address
  then
    set Button1.Visible to false
    set Button1.Visible to true

when Button2.Click
do
  call BluetoothClient1.Disconnect
  set Button1.Visible to true
  set Button1.Visible to false

when LocationSensor1.LocationChanged
do
  latitude longitude altitude
  set latitude to get latitude
  set longitude to get longitude
  set altitude to get altitude
  set global Utmalocalizacionconocida to LocationSensor1.CurrentAddress

when Canvas1.Touched
do
  call Canvas1.Clear

when Clock2.Timer
do
  set TextBox1.Text to

when Button3.Click
do
  close application
  
```

```

initialize global Variables to list from csv row text get start value
initialize global cuidador to 0
initialize global medico to 0
initialize global ejeprevio to 1
initialize global ejeX to 1
initialize global ejeY to 1
initialize global dato to 1
initialize global datoprevio to 1
initialize global fila to 0
initialize global Utmalocalizacionconocida to Desconocido
  
```

```

when Clock1.Timer
do
  if BluetoothClient1.IsConnected
  then
    call TextBox1.HideKeyboard
    set global dato to call BluetoothClient1.ReceiveUnassignedByteNumber
    set global dato to + get global dato + get global fila * 103
  or
    get global dato = 127
  or
    get global dato = 230
  or
    get global dato = 333
  or
    get global dato = 438
  then
    call Canvas1.Clear
    set Texting1.PhoneNumber to get global cuidador
    set Texting1.Message to Electrodo Desconectado
    call Texting1.SendMessage
  else if
    get global dato = 128
  or
    get global dato = 229
  or
    get global dato = 332
  or
    get global dato = 435
  then
    set LocationSensor1.Enabled to true
    set Texting1.PhoneNumber to get global cuidador
    set Texting1.Message to Evento Detectado. Enviar Correo al Doctor.
    call Texting1.SendMessage
    set Texting1.PhoneNumber to get global medico
    set Texting1.Message to join Evento Detectado. Mi ubicación es.
    get global UltimaLocalizacionconocida
    set global fila to 0
    call Texting1.SendMessage
    call TinyDB1.StoreValue
    tag
    valueToStore | call Canvas1.SaveAs
    fileName | hola.jpg
    call Sharing1.ShareFileWithMessage
    file
    message | /storage/emulated/0/hola.jpg
    Envío la señal del paciente
  
```

```

else
  call Canvas1 .Clear
  set global fila to 0
  set TextBox1 .Text to join
  get global dato
  call Canvas1 .DrawPoint
  x
  get global ejeX
  y
  get global dato
  call Canvas1 .DrawLine
  x1
  get global ejeXprevio
  y1
  get global datoprevio
  x2
  get global ejeX
  y2
  get global dato
  set global ejeX to remainder of ( get global ejeX + 1 ) ÷ 252
  if
  get global ejeX = 0
  then
  set global fila to modulo of ( get global fila + 1 ) ÷ 4
  set global dato to get global dato
  set global ejeXprevio to get global ejeX
  
```