Analog Circuits

EL 213

Winter 2019

Assigned By

Prof. Rutu Parekh

Group

10

Dhirubhai Ambani

Institute of Information and Communication Technology

Group Members :-

1) Rajan Ladani 201701029
2) Vedant Parikh 201701076
3) Jay Makwana 201701188
4) Harsh Patel 201701194
5) Dharmin Solanki 201701198
6) Ruchit Solanki 201701199
7) Jaymin Parmar 201701203
8) Om Palsanawala 201701216
9) Dhruv Patel 201701269
10) Mihir Patel 201701270

Table of Contents

Title

“To detect whether a person is dead or alive using BPM sensor (Pulse sensor) and Arduino Nano. And then transmit the message using ZigBee module to a PC.”

Abstract

The course EL 213 – Analog Circuits – as the name of the course says, mainly focuses on exploring the world of analog circuits consisting of various components. In theory part, analyzing complex circuits consisting transistors, capacitors, resistors, dependent power sources and such components makes us use our minds for critical thinking; whereas in project part, using different sensors, modules, micro-controllers and micro-processors, we need to apply our skills in the domain of electronics and wireless communication. Talking in particular, our project mainly aims on whether a person is alive or not. We've accomplished that using BPM sensor, also known as pulse sensor or heart rate sensor and processed using Arduino Nano and communicated using ZigBee module.

Acknowledgement

We begin by expressing our sincerest gratitude towards our course instructor, Prof. Rutu Parekh. Our interest in the subject - Analog Circuits - has been increasing exponentially, thanks to her and her exceptional views. We are highly indebted to Rutu Ma'am for guiding us continuously throughout the course of the semester and helping us in making this project a success. Her guidance has enhanced our capability in practical manners. We are greatly thankful to her for giving us such freedom to choose the topic that we like along with the freedom to explore the topic at our will and giving us a chance to showcase our creativity.

The group owes sincere thanks to her for providing us with this opportunity to explore something new. We would also like to thank the Teaching Assistants for guiding us throughout the project or in any kind of doubt for that matter. Last but not the least, we would like to thank the institution for providing us with the necessary infrastructure for the completion of this project.

Introduction

The project mainly aims at the detection of a person being alive or dead. We have used the Pulse sensor also known as BPM (Beats Per Minute) sensor or Heart Rate Sensor, which will count the pulse of the person, the data from the pulse sensor is processed by Arduino Nano and decided whether that person is dead or not depending upon his heart rate, and then that entire data, along with his heart-rate's wave form which is close to any normal cardio graph, is fed to the XBee module S2C (ZigBee), which will transmit it and the other XBee module which is at the receiver end, will receive it.

Now the obtained data, is sent to PC, which further needs MicroPython Console to decode it and XCTU software provides the inbuilt support for MicroPython Console.

Then the XCTU shows the data to the Personal Computer in the console log on the screen and it is shown whether that person is alive or not and also, in both the forms – hexadecimal and data that we want to print which we had sent from the transmitter.

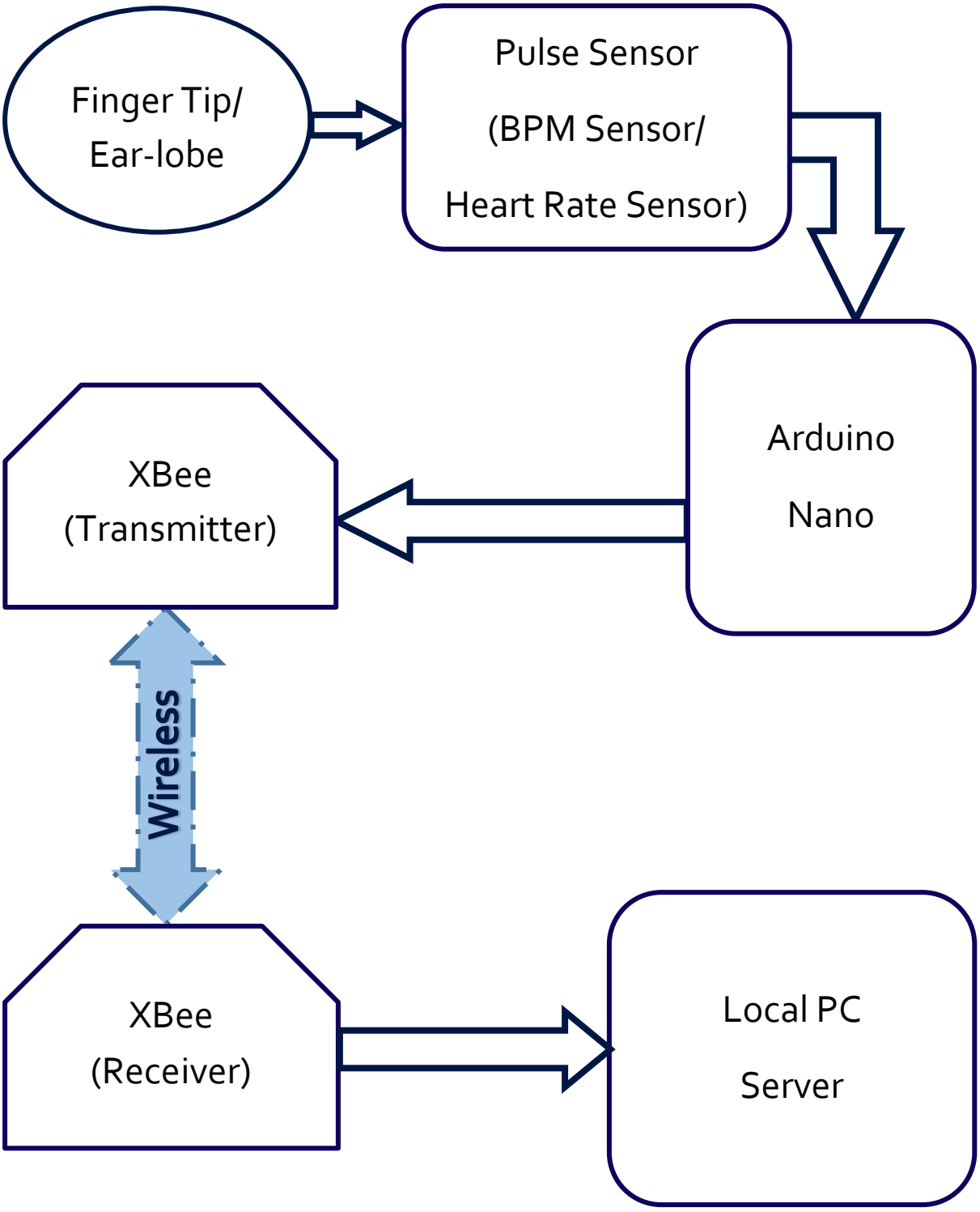
Project Objectives

The main objective of the project is to observe signal from pulse sensor and implement a device that analyzes and detects whether pulse sensor is touched by living thing or it is on non-living thing using Arduino Nano.

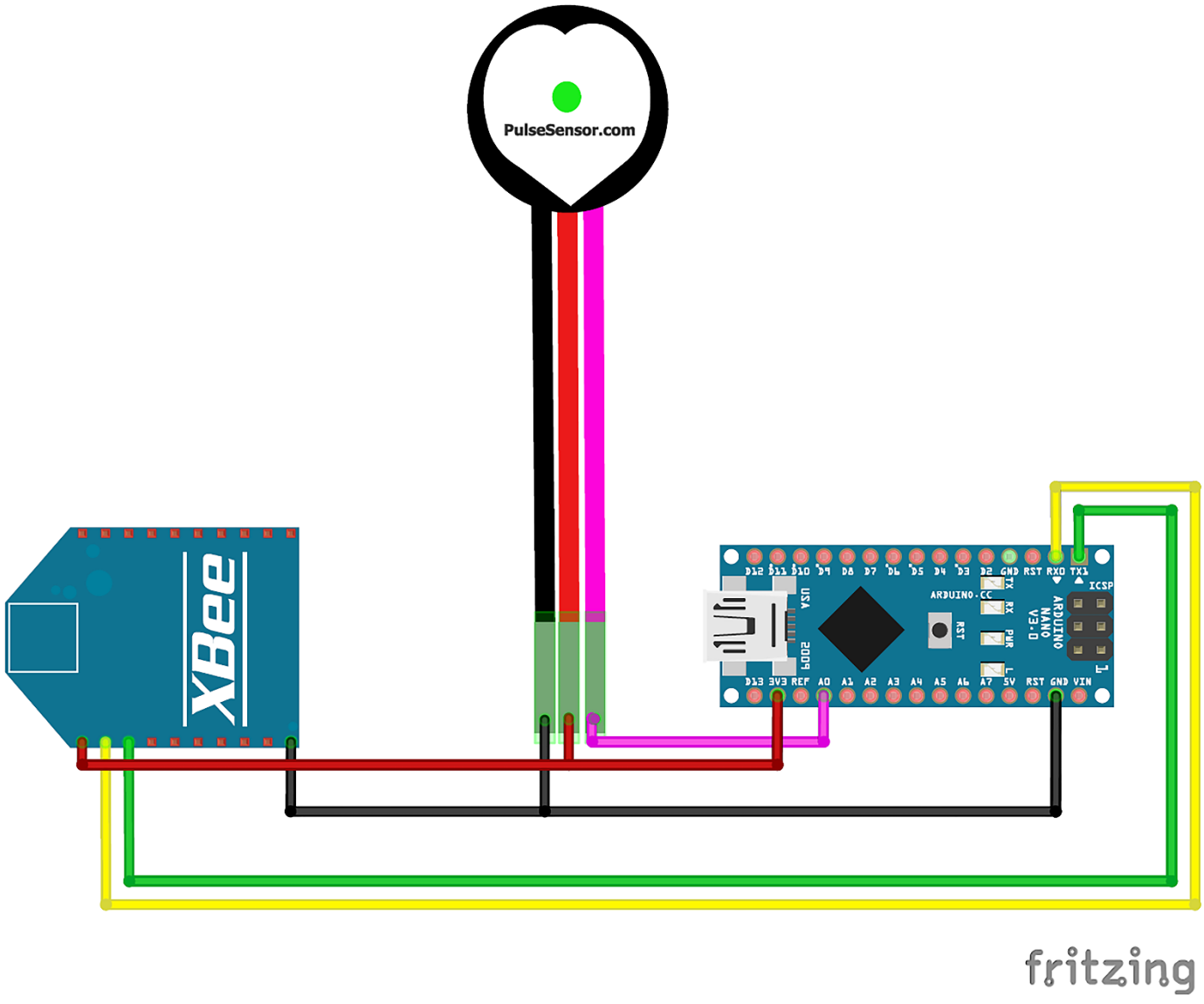
Purpose for implementing this device is that, using this device one can monitor whether a person is dead or alive and send the data about the status of that person from a remote location. The project can be further extended using more complex circuits and modules to show the accuracy and detect different types of death; for e.g. whether a person is brain-dead etc.

The objective for other users is that it's quite obvious that if there's any accident or other sort of casualty, the person should be directed straight to the hospital for further care. But when it's difficult to say whether person is still alive or not, this implemented project's working model- product can come very handy and useful.

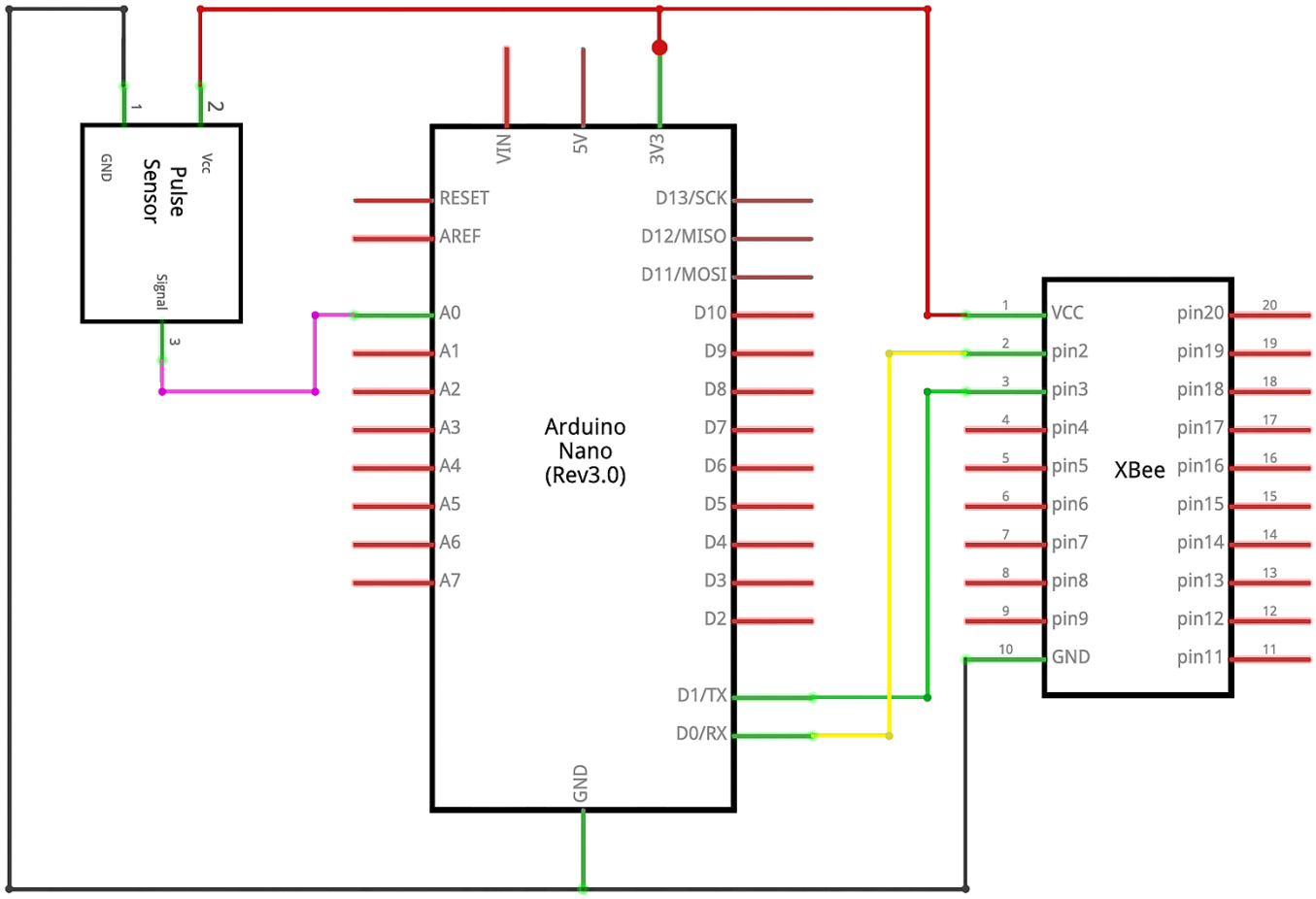
Block Diagram



Circuit Diagram



Circuit on Breadboard

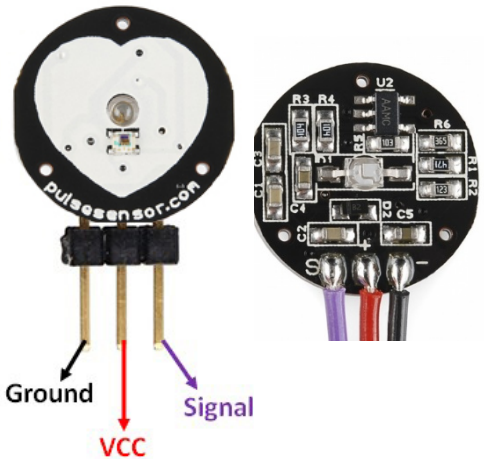


Circuit Schematic

P.S. : This circuit is at the transmitter side. At the receiver side, we just have XBee module connected to PC which doesn't require any sort of additional circuitry.

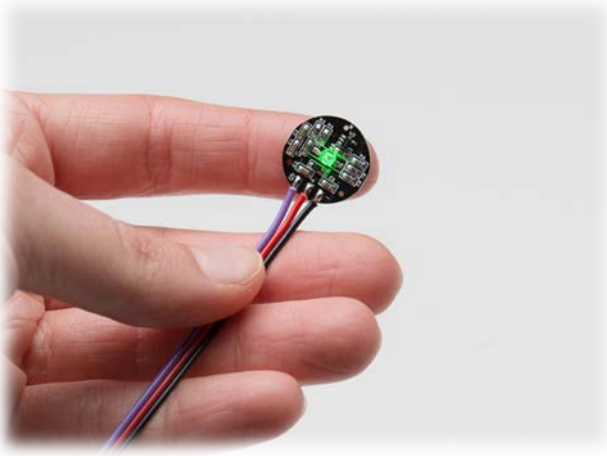
Components

1) BPM Sensor (Pulse Sensor – XD-58c) :



The Pulse sensor a.k.a. BPM sensor has two sides, on one side a green LED is placed along with an ambient light sensor connected with an LDR; and on the other side we have some circuit which consists of 6 resistors, 5 capacitors and LM324 ICs. This circuitry is responsible for the amplification and noise cancellation work. The pinout of the sensor is shown here.

Coming to how the sensor works, in order to detect the pulse, BPM sensor will emit light (using an LED) to the finger or earlobe or any other body-part which have capillary tissues of direct veins under the skin. Now whenever the heart pumps blood, more light is reflected by increased oxygenated blood cells and the ambient light sensor will observe this reflection of the intensity of light.



As a result, the resistance value of the LDR connected to it increases. This variation in resistance is converted into voltage variation using a signal conditioning circuit usually LM324 OP-Amp. The signal is amplified enough to be detectable by the microcontroller inputs. The micro-controller can be programmed to receive an interrupt for every pulse detected and count the number of interrupts

2) Arduino Nano :

As mentioned before, there has to be a microcontroller to process the data picked up by the sensor. We have used Arduino Nano. Arduino Nano is a microcontroller board designed by Arduino.cc. The microcontroller IC used in the Arduino Nano is Atmega328P, the same one as used in Arduino UNO. It has a wide range of applications and is a major microcontroller board because of its small size and flexibility.

Here are few of its basic features which you must know if you are thinking to work on this great microcontroller board:

- It has 22 input/output pins in total.
- 14 of these pins are digital pins.
- Arduino Nano has 8 analogue pins.
- It has 6 PWM pins among the digital pins.
- It has a crystal oscillator of 16MHz.
- Its operating voltage varies from 5V to 12V.
- It also supports different ways of communication, which are:
    - Serial Protocol.
    - I2C Protocol.
    - SPI Protocol.
- It also has a mini USB Pin which is used to upload code.
- It also has a Reset button on it.

3) ZigBee Module (XBee S2C) :

ZigBee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks (PANs) with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. In this personal area network, ZigBee defines three different device types: coordinator, router, and end device.

C Coordinator : ZigBee networks may only have a single coordinator device.

- This device starts the network, selecting the channel and PAN ID.
- This device manages the other functions that define the network, secure it, and keep it healthy.
- It cannot sleep; the coordinator must be powered on all the time.

R Router : A router is a full-featured ZigBee node.

- This device can join existing networks and send, receive, and route information. Routing involves acting as a messenger for communications between other devices that are too far apart to convey information on their own.
- It can allow other routers and end devices to join the network.
- It cannot sleep; router(s) must be powered on all the time.
- May have multiple router devices in a network.

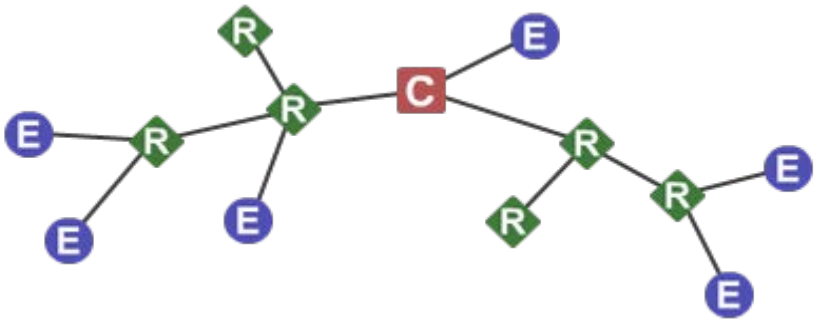
E End device : An end device is essentially a reduced version of a router.

- This device can join existing networks and send and receive information, but cannot act as messenger between any other devices.
- It cannot allow other devices to join the network.
- It uses less expensive hardware and can power itself down intermittently, saving energy by temporarily entering a non-responsive sleep mode.
- It always needs a router or the coordinator to be its parent device. The parent helps end devices join the network, and stores messages for them when they are asleep.
- ZigBee networks may have any number of end devices.

In fact, a network can be composed of one coordinator, multiple end devices, and zero routers. To configure any module into any of these three devices, a XBee USB Explorer is used to mount the module and further, can be connected to a PC and configured via a software, XCTU.

XBee USB Explorer

The following diagram shows a generic ZigBee network.



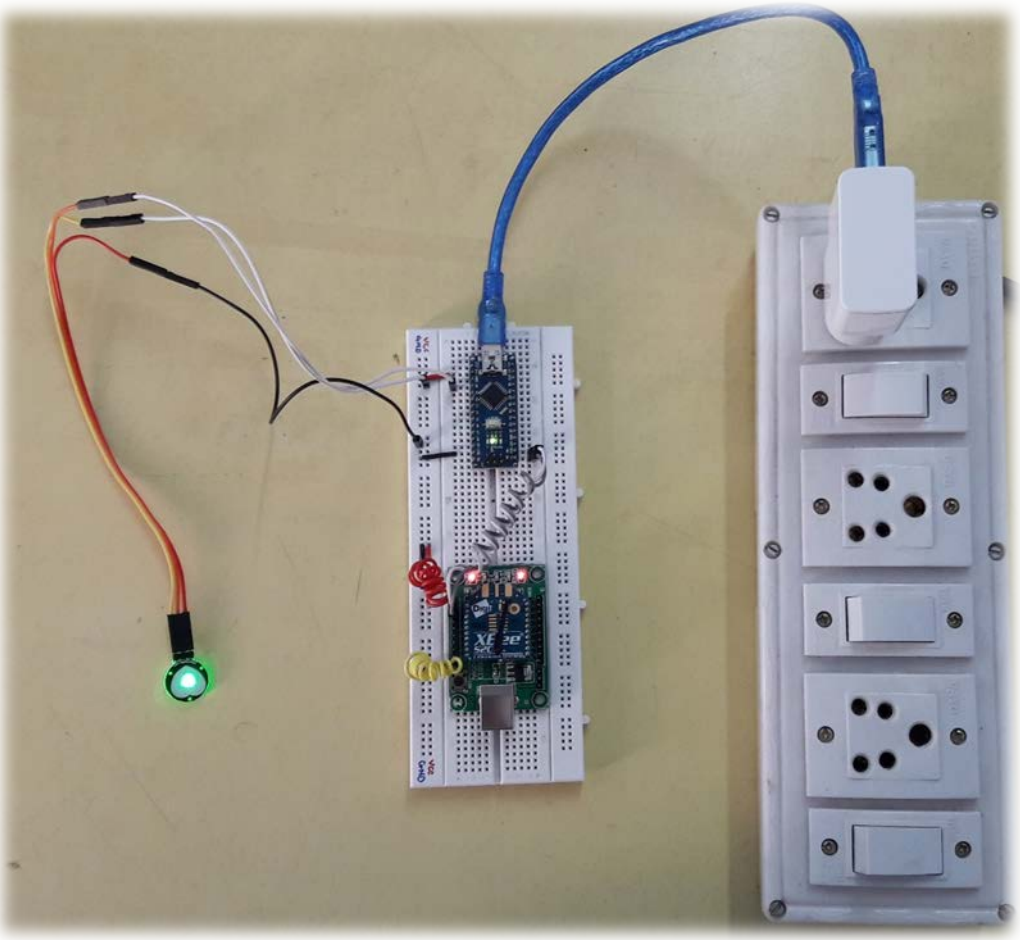
Note : Each ZigBee network must be formed by one, and only one, coordinator and at least one other device (router or end device).

A device can only operate as one of the three device types. The device type is selected by configuration rather than by firmware image as was the case on earlier hardware platforms.

By default, the device operates as a router. To select coordinator operation, set CE to 1. To select end device operation, set SM to a non-zero value. To select router operation, both CE and SM must be 0.

Description and Working of Model

The circuitry is connected as shown in the circuit diagram. Vcc of both – BPM sensor and XBee S2C (ZigBee) module is connected to Arduino Nano's 3V3 pin and ground pin of both – BPM sensor and XBee S2C (ZigBee) module is connected to Arduino Nano's GND pin. The Data_Out pin (Transmitter-Tx pin) of ZigBee is connected to the Rx (Receiver pin) of Arduino Nano and the Data_In pin (Receiver-Rx pin) of ZigBee is connected to the Tx (Transmitter pin) of Arduino Nano.



We first give input through BPM sensor by putting finger on it. We can also put the sensor on our earlobe. The backside circuitry of BPM sensor is coated by insulator material (hot melt glue) to avoid interference of noise as the circuit was open. BPM sensor has three pins.

First pin is for ground, second one is for Vcc and third is for passing output of BPM sensor to Arduino Nano – the microcontroller board that we used.

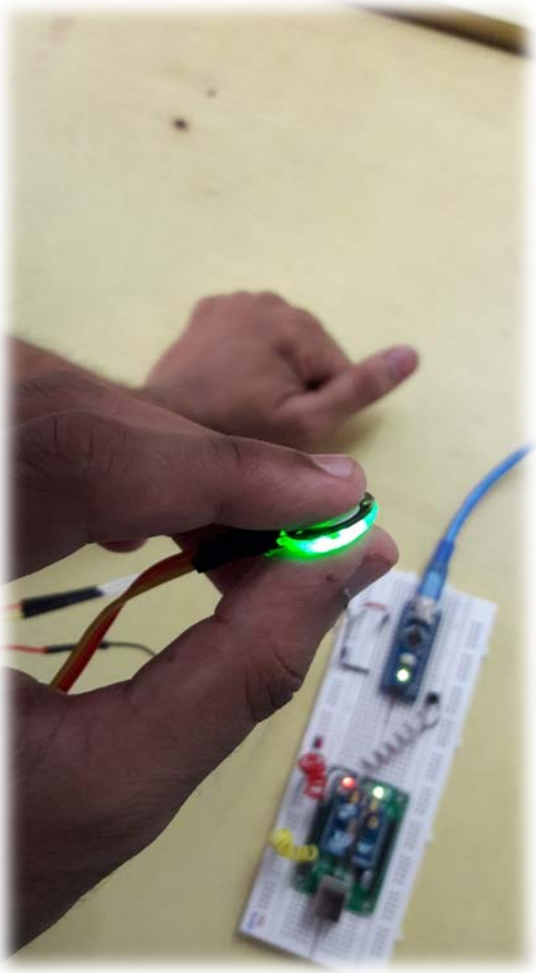
We give external supply voltage to Arduino Nano through adapter. Arduino can convert this voltage to 5V or 3.3V using the inbuilt voltage regulator ICs, from which we give 3.3V to BPM sensor as Vcc to power up the circuit. We connect ground pin of BPM sensor to ground of Arduino. Output of BPM sensor is given to microcontroller through its analog pin. Output of BPM sensor is in range of 0 to 700. After analyzing several outputs, trial and error method, we came to conclusion that if signal output is greater than 550 which is threshold value for BPM xd-58c, then we show it as disconnected.

When the BPM sensor is suddenly connected or get in touch with something living or non-living, it will give output below 10. The above 2 observations dictate that the Sensor value is dependent upon intensity of incident light-ray and reflected ray. Signal value might be subtraction of those two intensities. When BPM sensor is connected to any non-living thing, it will show constant value around 340 as there is constant difference between incident and reflected rays. When the sensor is connected to living body, it will show value greater than 340 and varies frequently. This all happens in background. The console log will only show you whether you're dead or alive!

Anyway, after that, we have to send these data to the computer using XBee module. For these we use two XBee S2C. One XBee is connected to Arduino and another one is to computer with the use of UNO cable and Explorer. For data transfer, we need to establish a network between two XBee. For this we first need to configure them by setting up the parameters accordingly with the help of XCTU. To transfer the output of Arduino we connect Tx pin of Arduino to Rx pin of XBee. After that to establish wireless network between two XBee, we set the 'MY' and 'DL' parameters of them. After this we will be able to see the output data in computer which is transferred by second XBee to computer.

Now, if the BPM sensor is not connected to anything, it'll show 'Disconnected' in the console log as output. Now if you're holding it, for the initial time it'll show 'Connecting...'. Once it's connected, it'll show the status (alive or dead). If you're not holding it properly, there'll be a message shown 'Please hold it properly...'. And you're done!

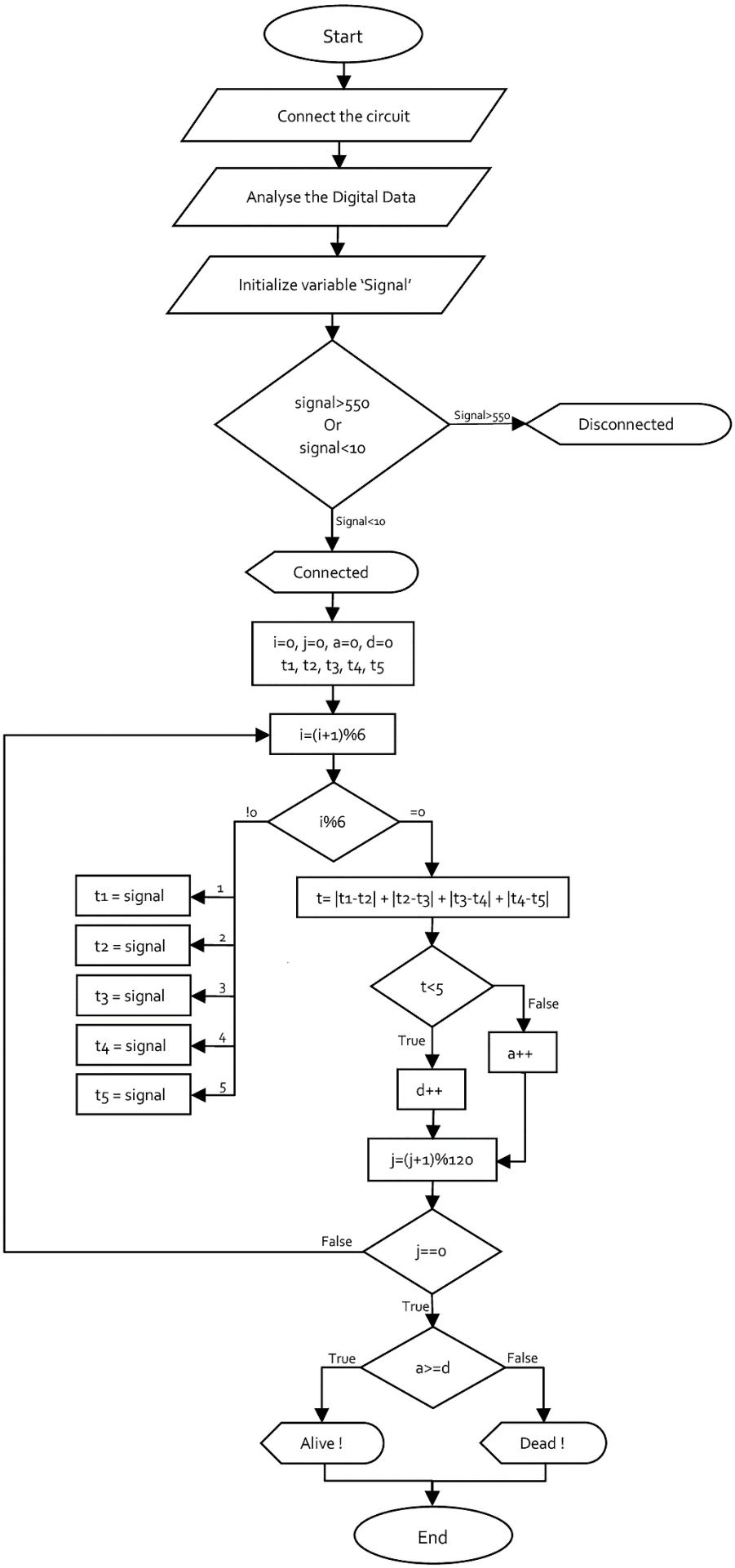
It's a simple pulse sensor and that's why it is not always accurate. So when you hold, at first, there might be displaying of dead instead of alive or somewhat same, so have patience and hold it for a while and you'll get the data that you want. Still, when you put finger, there might be the case it'll never detect you as alive despite you are alive! So, for the best results, it's advised to put the sensor on earlobe instead of finger.



Flow Chart

- From Bpm's analog pin we get analog signal of beat.
- We get Digital form of this signal on Serial monitor.
- Signal is variable that collect digital samples (at 50 ms delay).
- 550=Threshold value of signal.
- When signal <550 && signal >10. If the value of signal remain constant then detected **Dead!** Or if it varies around Avg. value then Detected **Alive!**
- d=dead counts
- a=alive counts
- i=counter which count the 5 samples
- t1, t2, t3, t4, t5=store the 5 samples
- t=Calculate Avg. variance of this samples.

These are some terms to be understood before further moving on to the flow chart.


Start
Connect the circuit
Analyse the Digital Data
Initialize variable 'Signal'
signal>550
Or
signal<10
Signal>550
Disconnected
Signal<10
Connected
i=0, j=0, a=0, d=0
t1, t2, t3, t4, t5
i=(i+1)%6
i%6
!0
=0
t1 = signal
t2 = signal
t3 = signal
t4 = signal
t5 = signal
1
2
3
4
5
t= |t1-t2| + |t2-t3| + |t3-t4| + |t4-t5|
t<5
False
True
a++
d++
j=(j+1)%120
False
j==0
True
True
a>=d
False
Alive !
Dead !
End

Test Results

Check for different things-

1. **Disconnected:** Just leave the sensor. First it will show values around 640 for which will help us determining the status (dead or alive) Disconnected after some 2,3 seconds noise adds and it automates the values to be alive so, values shown immediately after disconnected is redundant.

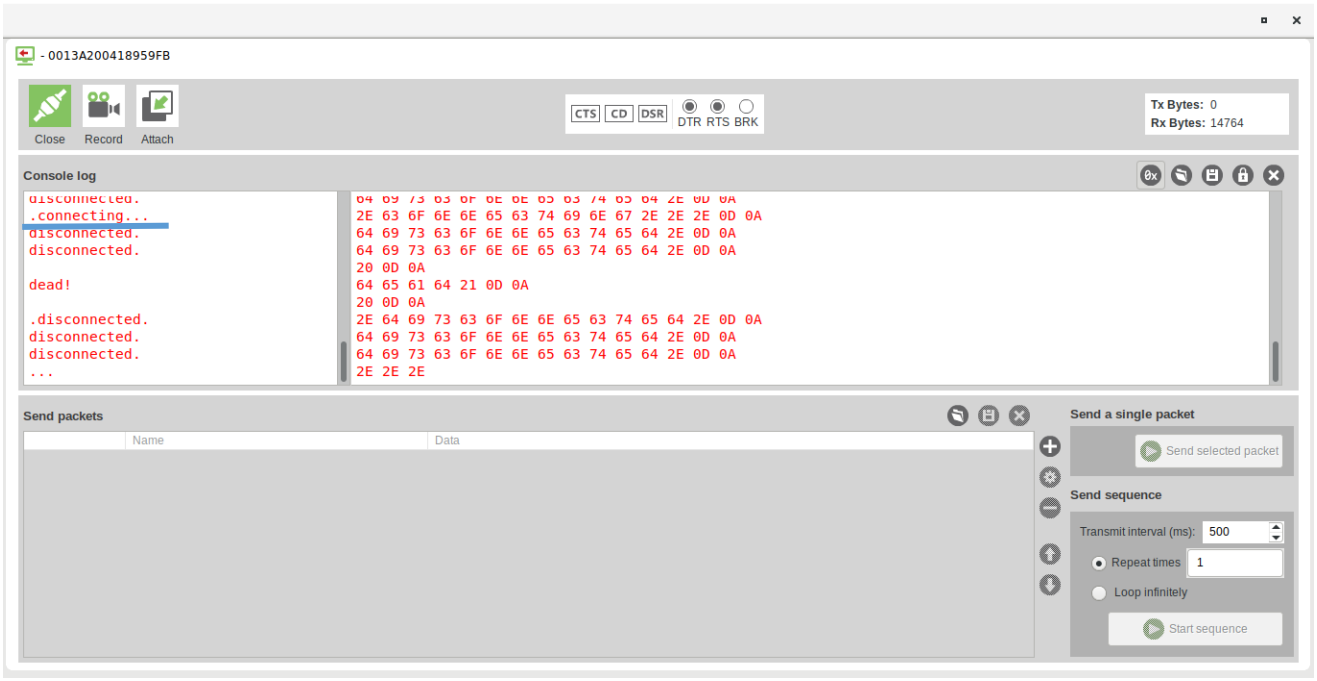
2. **Connecting**: It shows signal values less than 10. So, we have connected our sensor to something. Values after connecting is considerable.

3. **Alive:** Hold Properly. Not very tight, and not very loose and signal data is around 360 and it varies greater than 5 or 7 in most of the cases. So, a bit increment means we counted 5 consecutive signal values as alive.

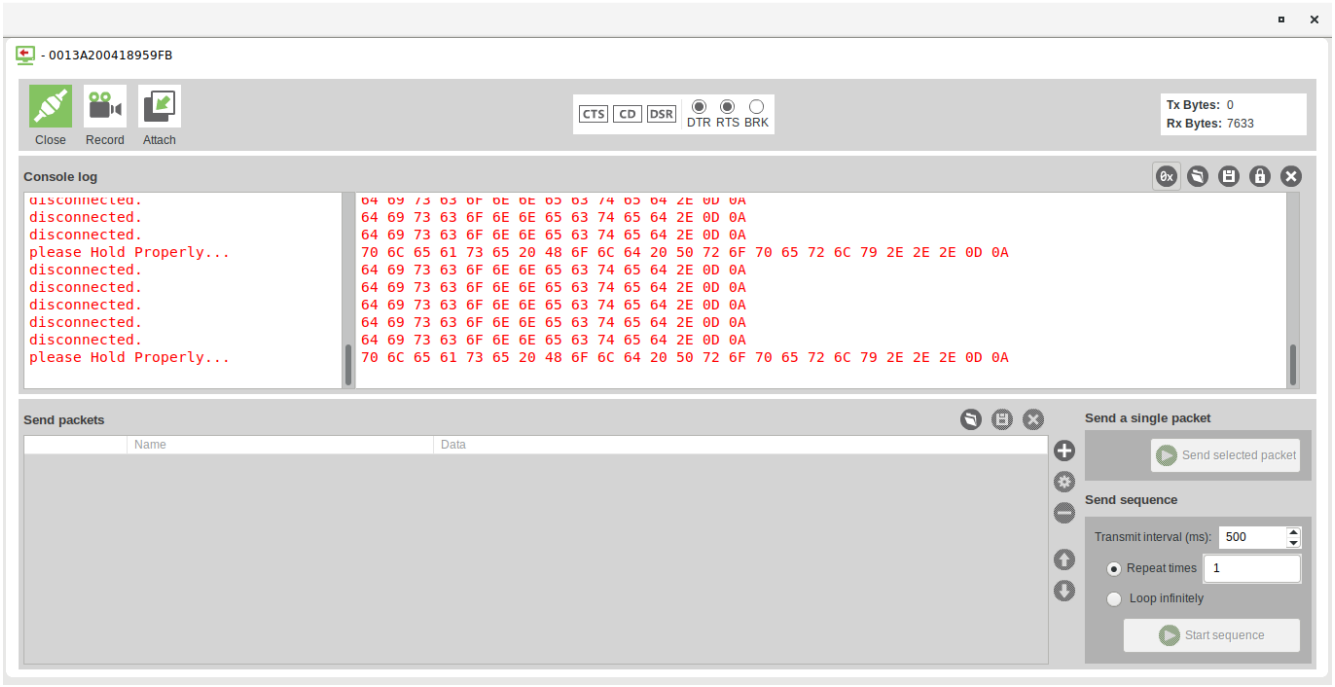
4. **Dead:** Touch the BPM sensor to a non-living thing. Signal values shows not much variance, why? Because no noise or no absorption of light is there. So, it varies no more than 4, mostly 0,1,0,1 .... Variation will be shown. We count 5 consecutive bits as dead. After 3 seconds, if counts are greater than d counts then the organism is alive else it's dead.

After all the process, we have observed that our project is 70% accurate. Due to some reasons, 30% of the time it shows different data. As mentioned above, we need to have patience and use earlobe for the best and accurate results.

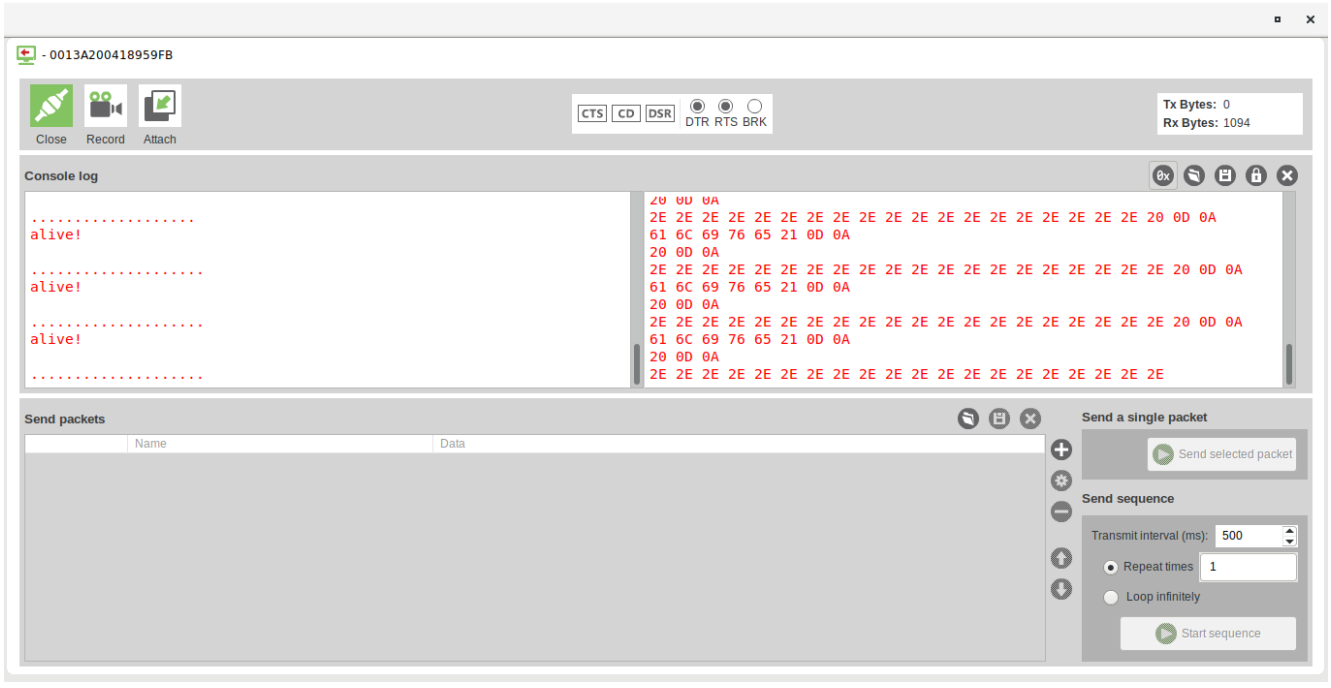
When it was disconnected



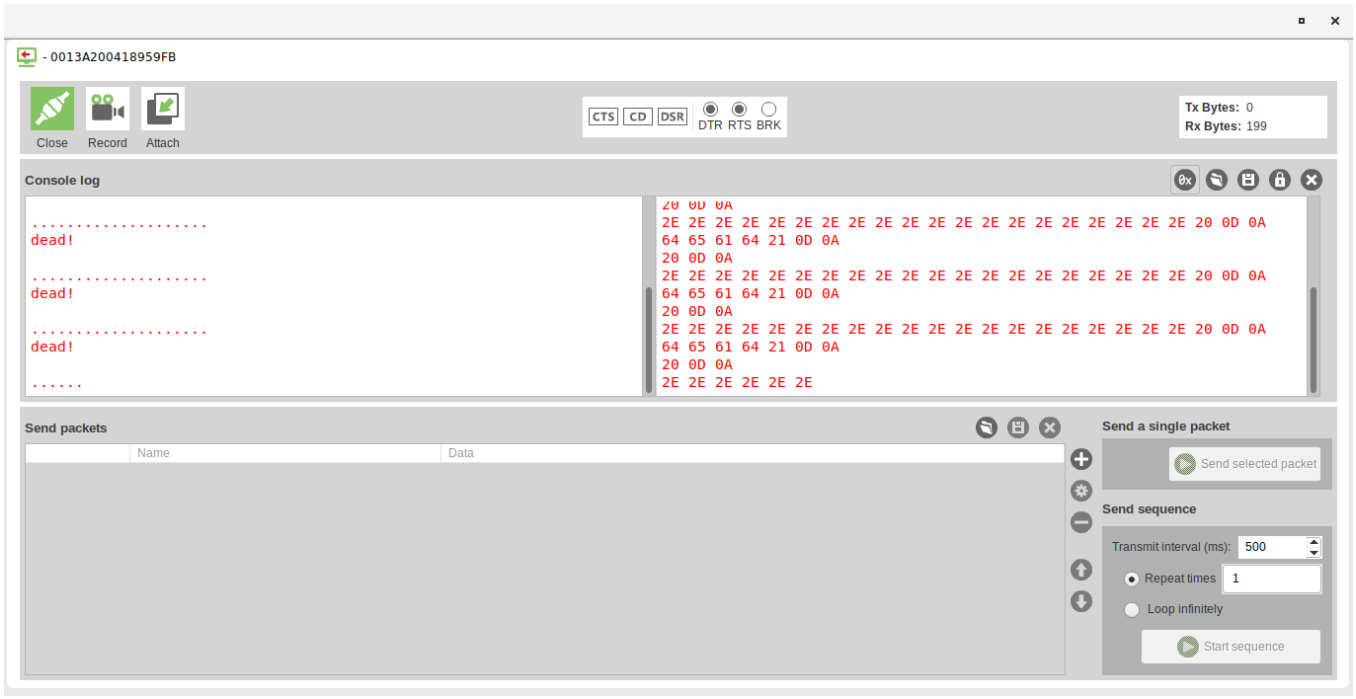
When it was connecting



When it was not held properly



When it was held by finger or earlobe



When it was kept attached to a non-living thing

Conclusion

The project successfully detected whether the person is alive or dead, with the help of the pulse sensor, Arduino Nano and XBee. What we can do as an improvement to this project is to detect if a person is alive and if only a particular organ of the person is not functioning. This can be achieved with the help of some modifications. Also, apart from detecting whether a person is dead or alive, the pulse sensor can also be used to design a system which detects the heart rate. Heart rate data can be really useful whether you're designing an exercise routine, studying your activity or anxiety levels or just want your shirt to blink with your heart beat. The problem is that heart rate can be difficult to measure. Luckily, the Pulse Sensor can solve that problem!

References

- https://www.elprocus.com/heartbeat-sensor-working-application/
- https://www.mepits.com/project/299/embedded-projects/zigbee-arduino-interfacing
- https://www.elprocus.com/what-is-zigbee-technology-architecture-and-its-applications/
- https://alselectro.wordpress.com/2017/01/23/zigbee-xbee-s2c-how-to-configure-as-_coordinator-router-end-device/
- https://www.digi.com/pdf/chart_xbee_rf_features.pdf/
- https://www.elprocus.com/arduino-basics-and-design/
- https://www.youtube.com/watch?v=ABW02F6c80A
- https://www.how2electronics.com/heartbeat-pulse-bpm-rate-monitor- using-arduino-pulse-sensor/
- https://www.rs-online.com/designspark/basics-of-arduino-nano
- https://www.youtube.com/watch?v=02fNHdCxCD0
- https://ieeexplore.ieee.org/document/7181913
- https://www.raviyp.com/embedded/140-learn-how-a-heart-beat-sensor-works

- https://www.rs-online.com/designspark/basics-of-arduino-nano
- https://en.wikipedia.org/wiki/Zigbee
- https://www.digi.com/resources/documentation/Digidocs/90002002/Content/Concepts/c_device_types.htm?TocPath=zigbee%20networks%7Czigbee%20networking%20concepts%7C_____1
- https://pulsesensor.com/
- https://www.digi.com/products/iot-platform/xctu
- https://github.com/WorldFamousElectronics/PulseSensorPlayground/blob/master/resources/PulseSensor%20Playground%20Tools.md
- https://www.generationrobots.com/media/DetecteurDePoulsAmplifie/PulseSensorAmpedGettingStartedGuide.pdf
- https://learn.sparkfun.com/tutorials/exploring-xbees-a
- https://learn.sparkfun.com/tutorials/exploring-xbees-and-xctu/all
- https://learn.sparkfun.com/tutorials/xbee-shield-hookup-guide/all

Thank You...!!