ASSIGNMENT 2 FRONT SHEET

Qualification	**TEC Level 5 HND Diploma in Computing**		
Unit number and title	**Unit 43: Internet of Things**		
Submission date	April 24, 2021	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Luu Vinh Truc Phuong	**Student ID**	GCD191122
Class	GCD0806	**Assessor name**	Hoang Nhu Vinh
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	 Luu Vinh Truc Phuong

Grading grid

P5	P6	P7	M5	M6	D2	D3

❒ Summative Feedback:	❒ Resubmission Feedback:

Grade:	Assessor Signature:	Date:

Internal Verifier's Comments:

Signature & Date:

I. Overview Product....1

1. Model product....1

1.1. NodeMCU....1

1.2. MAX30100 sensor....3

1.3. Blynk....6

1.4. Arduino IDE....14

2. Design....17

2.1. Schematic....17

2.2. Specification....17

2.3. Sketch....18

3. Code and implement....19

4. Demo project....23

4.1. Testing....23

5. Link Video:....27

II. Review & Feedback....28

1. Review....28

2. Evaluate end user feedback from your IoT application....30

2.1. Evaluate usability and commercialization....30

2.2. Feasibility evaluation....30

2.3. Ability to improve....31

3. Development plan....31

III. References....35

INDEX OF COMMENTS....36

TABLE OF FIGURES

Figure 1: Model Product.....1
Figure 2: NodeMCU.....2
Figure 3: NodeMCU pin structure.....3
Figure 4: MAX30100 sensor.....4
Figure 5: System Block Diagram.....5
Figure 6: Pin Configuration of MAX30100.....5
Figure 7: Pin Description of MAX30100.....6
Figure 8: Widget Box.....7
Figure 9: Blynk connectivity protocol.....8
Figure 10: Login Interfaces.....9
Figure 11: Blynk Interfaces.....10
Figure 12: Choose the hardware.....11
Figure 13: Author Token.....12
Figure 14: Widget Box.....13
Figure 15: Notification setting.....14
Figure 16: Preferences UI.....15
Figure 17: Board Manager.....15
Figure 18: Library Manager.....16
Figure 19: Schematic.....17
Figure 20: Blynk installation.....18
Figure 21: Code and Implement.....23
Figure 22: Test cases.....27
Figure 23: Multiple-choice question.....28
Figure 24: Project Detector Comments.....29
Figure 25: Good and bad points.....29
Figure 26:The column chart.....30

TABLE OF TABLES

Table 1: Circuit Pin structure.....17

Table 2: Project Specification.....18

Table 3: Development Plan.....34

I.

II. Overview Product

1. Model product

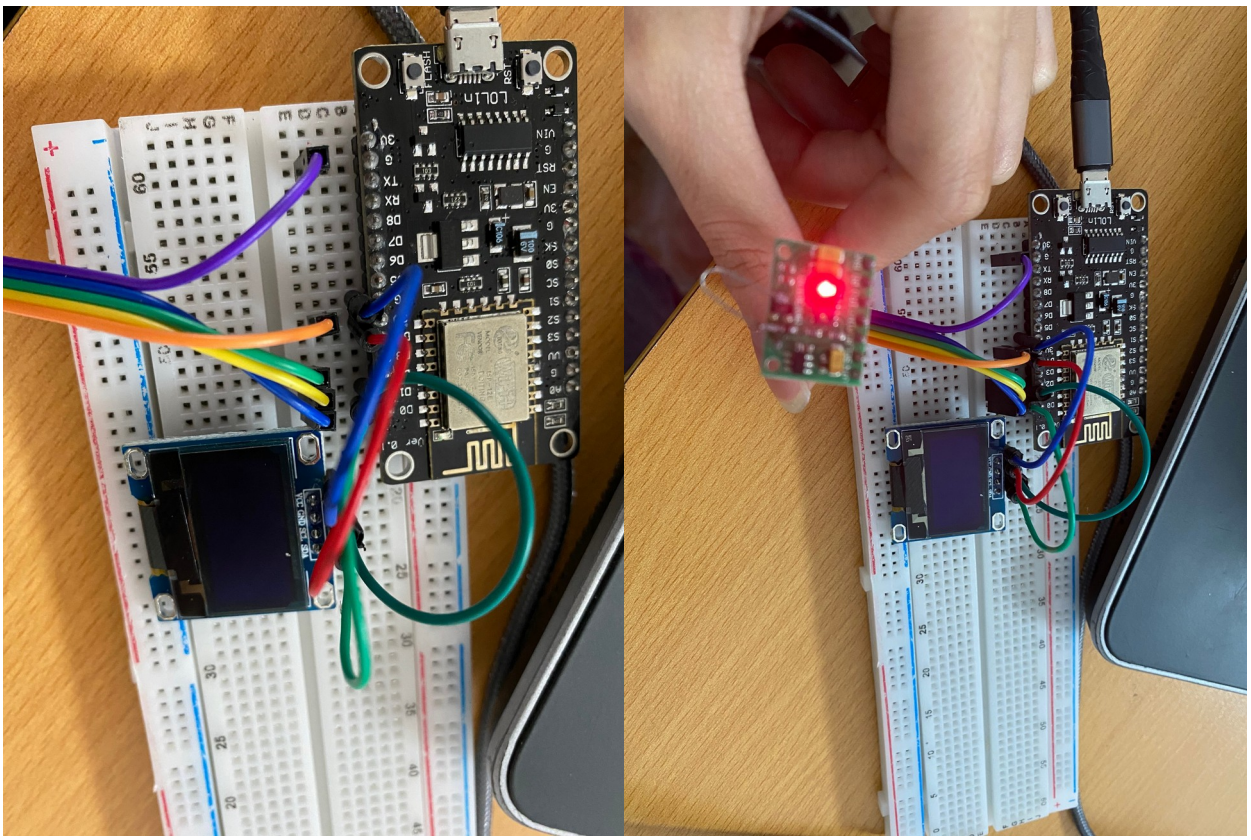


Figure 1: Model Product

In this product, I use some devices:

1.1. NodeMCU

NodeMCU is a very small compact board capable of programming and connecting to the network over a wireless network. In other words, the NodeMCU board is an on-chip system (SOC) with built-in TCP / IP protocol. In a network, the NodeMCU can act as a server, a client, or both. In server mode, NodeMCU applies as the host and in client mode NodeMCU can make requests on the server. In addition, the NodeMCU has input / output pins that can be connected to sensors or actuators so that data from the

sensor can be sent to the server and can trigger actuators based on the data, whether received. [CITATION Tin201 \l 1033]

Actually, general purpose input / output (GPIO) is one pin on the IC (Integrated Circuits). It can be either input or output pins, which can control their behavior at runtime. NodeMCU Development Kit provides access to these ESP8266 GPIOs. The only thing to note is that the NodeMCU Devkit pins are numbered differently from the internal GPIO notation of the ESP8266 as shown in the figure and table below. For example, D0 pin on NodeMCU Devkit is mapped to internal GPIO 16 pin of ESP8266.

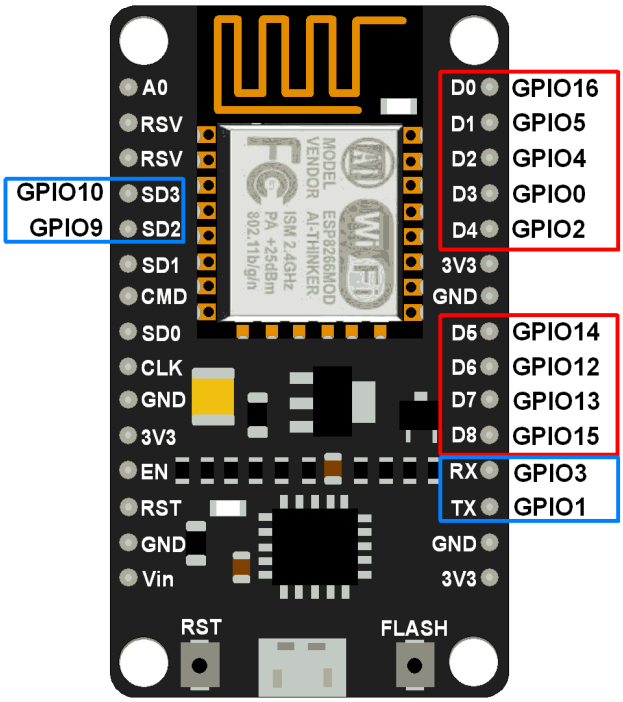


Figure 2: NodeMCU

The table below provides the NodeMCU Dev Kit IO pins and mapping the internal GPIO pins of the ESP8266

Pin Names on NodeMCU Development Kit ESP8266 Internal GPIO Pin number	Pin Names on NodeMCU Development Kit ESP8266 Internal GPIO Pin number
D0	GPIO16
D1	SCL/GPIO5
D2	SDA/GPIO4
D3	GPIO0
D4	GPIO2

D5	GPIO14
D6	GPIO12
D7	GPIO13
D8	GPIO14
D9/RX	GPIO3
D10/TX	GPIO1
D11/SD2	GPIO9
D12/SD3	GPIO10

Figure 3: NodeMCU pin structure

1.2. MAX30100 sensor

30100 is an integrated solution to measure oxygen pulse rate and heart rate monitoring. It combines two LEDs, one optimized optical, optical splitter and low-noise analogue signal processing to detect pulse oximetry and heart rate measurement signs. [CITATION EVA14 \l 1033]

The MAX30100 operates from 1.8V and 3.3V power supply and can be powered off via software with negligible backup current, allowing for power supply always connected.

Figure 4: MAX30100 sensor

Benefits and features:

- Complete pulse oxygen meter and heart rate sensor simplified solution for design
  - Integrated LED Light, High Performance Analog Front and Image Sensor – End
  - Tiny 5.6mm x 2.8mm x 1.2mm 14 pins system in advanced package
- Super energy-saving operation to increase battery life Wearable equipment
  - Programmable sampling rate and LED current for energy saving
  - Extremely low off current (0.7µA, typ)
- Advanced function Improve measurement efficiency
  - High SNR provides powerful motion artifact recovery
  - Integrated ambient light suppression
  - High sampling capacity
  - Fast data output capability

System block diagram:

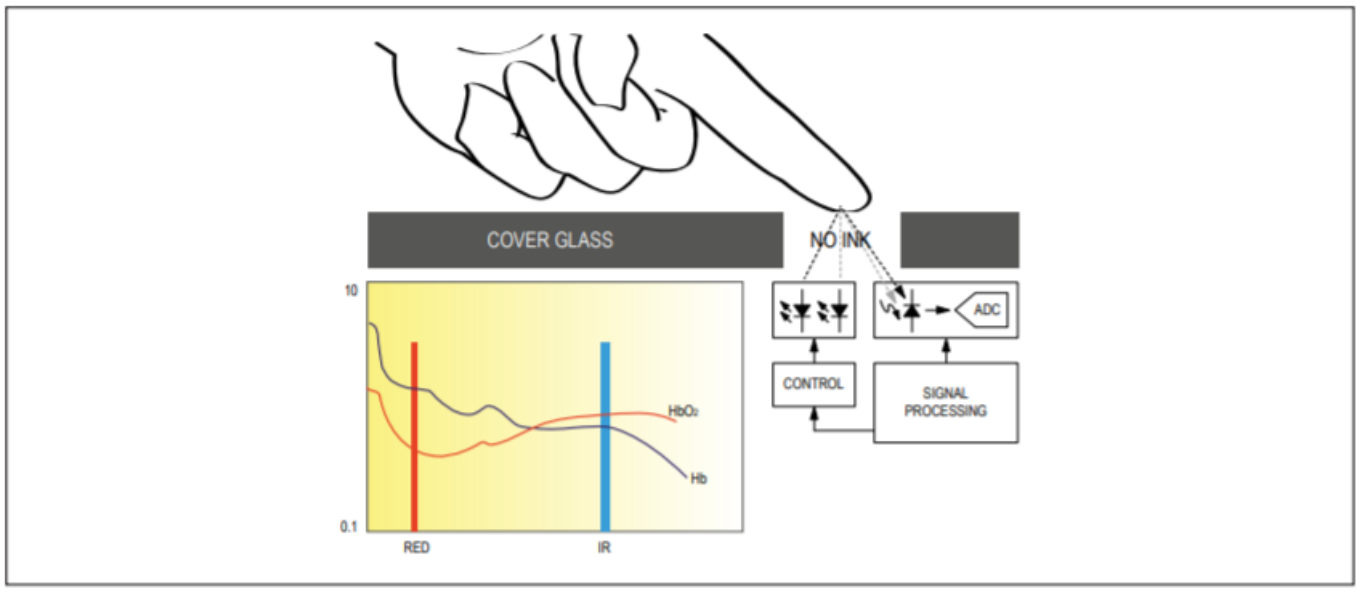


Figure 5: System Block Diagram

Pin configuration:

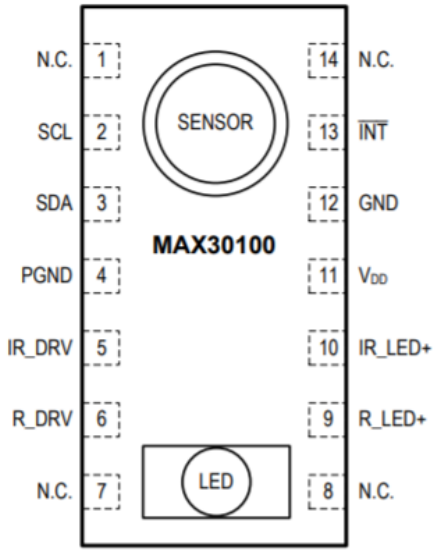


Figure 6: Pin Configuration of MAX30100

Pin Description:

PIN	NAME	FUNCTION
1, 7, 8, 14	N.C.	No Connection. Connect to PCB Pad for Mechanical Stability.
2	SCL	I²C Clock Input
3	SDA	I²C Clock Data, Bidirectional (Open-Drain)
4	PGND	Power Ground of the LED Driver Blocks
5	IR_DRV	IR LED Cathode and LED Driver Connection Point. Leave floating in circuit.
6	R_DRV	Red LED Cathode and LED Driver Connection Point. Leave floating in circuit.
9	R_LED+	Power Supply (Anode Connection) for Red LED. Bypass to PGND for best performance. Connected to IR_LED+ internally.
10	IR_LED+	Power Supply (Anode Connection) for IR LED. Bypass to PGND for best performance. Connected to R_LED+ internally.
11	V_{DD}	Analog Power Supply Input. Bypass to GND for best performance.
12	GND	Analog Ground
13	$\overline{INT}$	Active-Low Interrupt (Open-Drain)

Figure 7: Pin Description of MAX30100

1.3. Blynk

As I mentioned in asm1, Blynk is designed for the Internet of Things. It can receive and retrieve digital data sets from the accelerometer and gyroscope for analysis.

In this section, I will explain the three main components that are present in the background:

- Blynk App - allows creating skins for our products by dragging and dropping various widgets that the vendor has designed. In Heart Rate and Blood Oxygen Concentration, I need a component that will generate a notification and pass it on to the user via their smartphone, so the widgets I used are in the Notifications section inside the Convenience Box is useful in the image below.

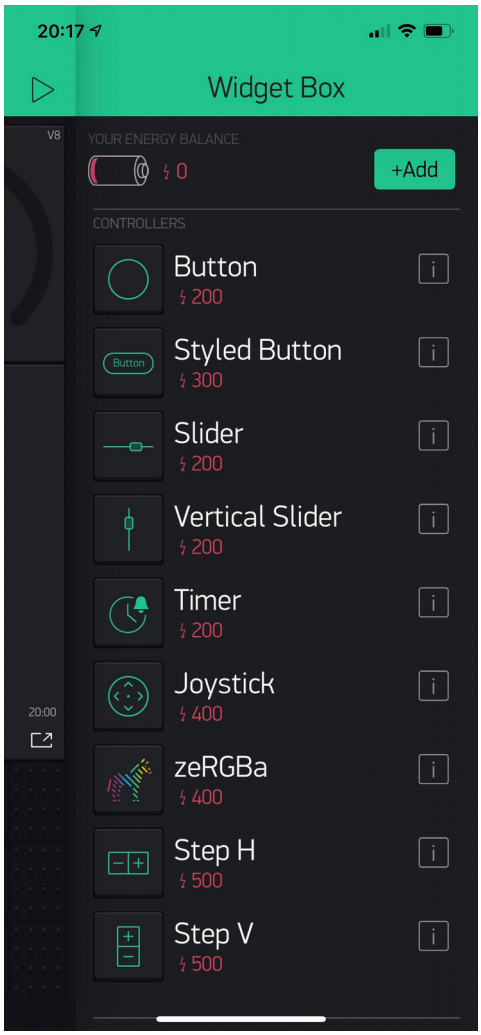


Figure 8: Widget Box

- Blynk Server - responsible for central data processing between phone, tablet and hardware. We can either use Blynk's Blynk Cloud or create our own Blynk server. Since this is open source, we can easily integrate the devices and even use NodeMCU as the server.
- The Blynk library - which supports most popular hardware platforms - allows communication with the server and handles all incoming and outgoing commands.

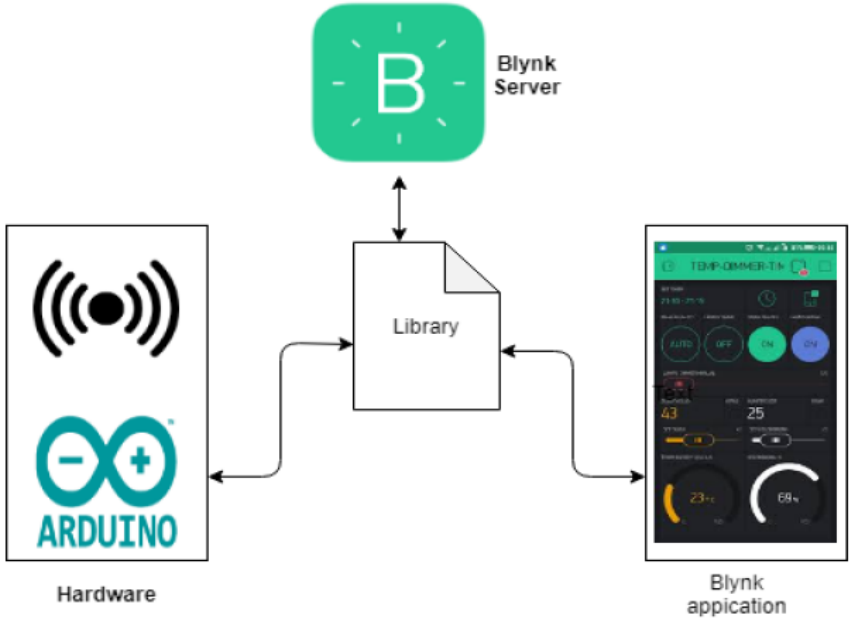


Figure 9: Blynk connectivity protocol

⇨ Now imagine: every time you press a button in the Blynk application, the request goes to Blynk's server, the server will connect to your hardware via the library. Similarly, the hardware device transmits the data back to the server

Features, characteristics:

- Provides the same API and UI for all supported devices and hardware
- Connect to the server by:
  - Wi-Fi
  - Bluetooth and BLE
  - Ethernet
  - USB (Serial)
  - GSM
  - ...
- The utilities on the supplier's interface are very easy to use
- Drag and drop interface directly without coding
- Easily integrate and add new functionality with the virtual connection ports built into the Blynk application

- Track historical data
- Communicate from device to device using Widgets
- Send emails, tweet, real-time notifications, and more.

How do I apply Blynk to my system?

Step 1: Create a Blynk account

After downloading the App Blynk successfully, we need to create a New Blynk account. This account is separate from the accounts used for the Blynk Forum. We recommend using a real email address as it will simplify things late

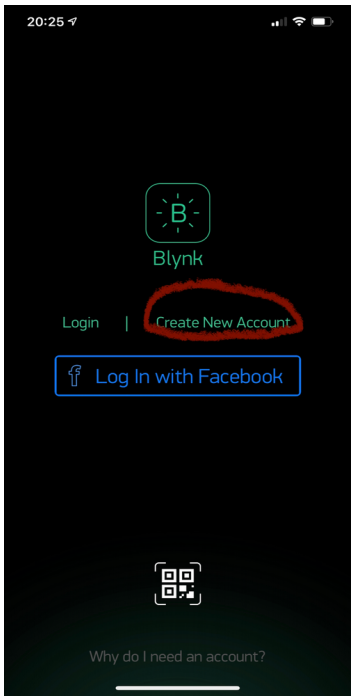


Figure 10: Login Interfaces

Step 2: Create a new project

Once you've successfully logged in to your account, start by creating a new project.

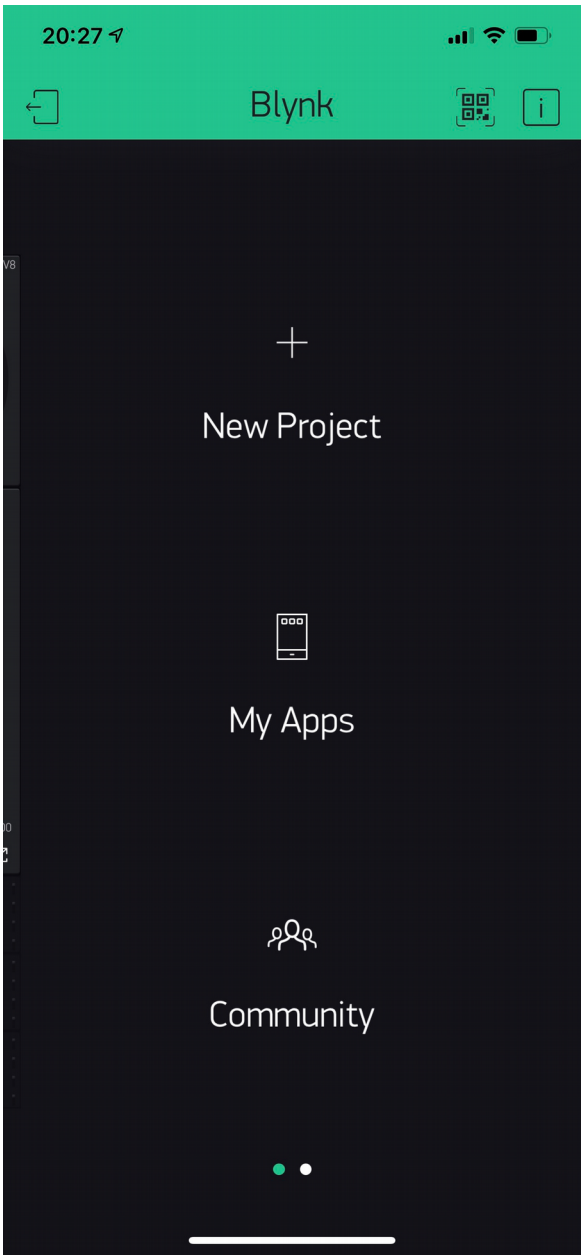


Figure 11: Blynk Interfaces

Step 3: Choose the hardware

Select the type of hardware you will use. Check the list of supported hardware and connection type.

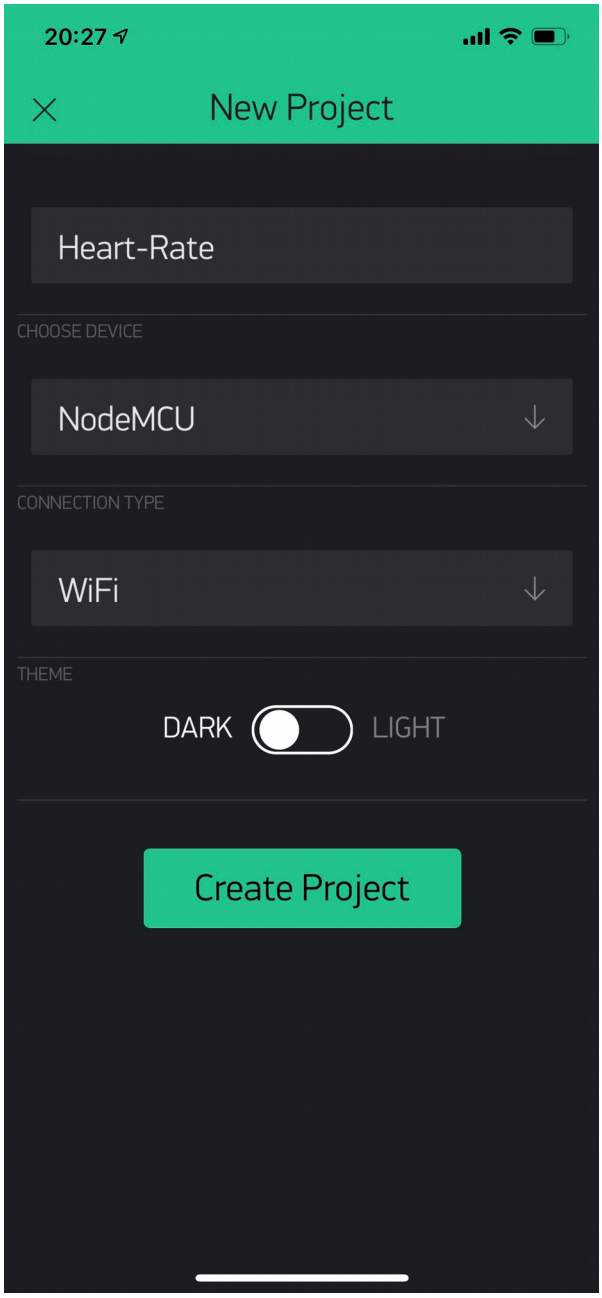


Figure 12: Choose the hardware

Step 4: Author Token

The authentication code is a unique identifier needed to connect a hardware to your smartphone. Each new project you create will have its own Authentication Code. You will automatically receive the Authentication Code on your email after creating your project, you can also copy it manually. You will see the token in:

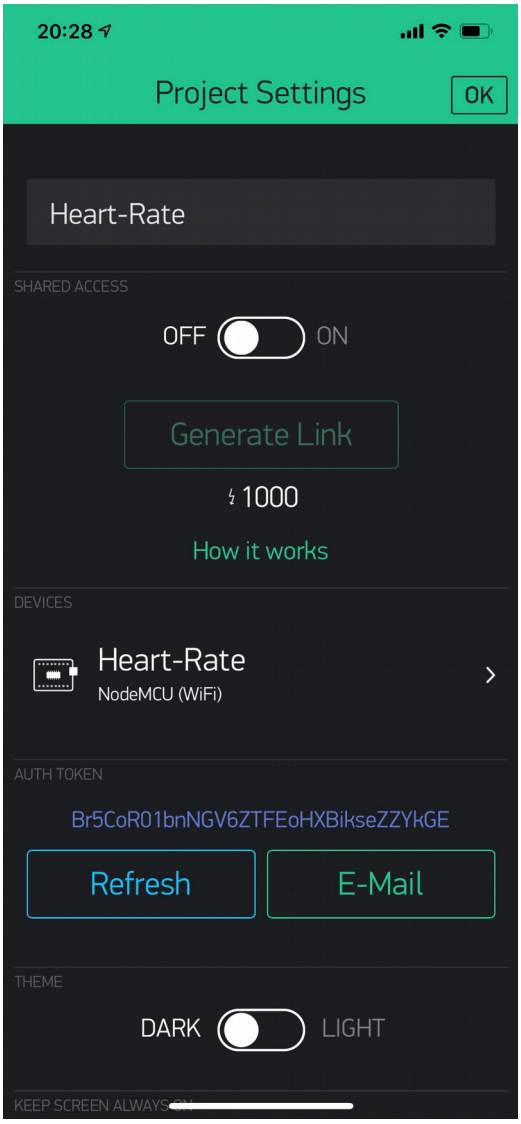


Figure 13: Author Token

Very convenient to send it by e-mail. Press the e-mail button and a token will be sent to the e-mail address you used to sign up. You can also tap on the Token line and it will be copied to the clipboard. Finally, press the "Create" button.

Step 5: Add widgets

Your project frame is empty, add a two Gauge and one SuperChart button to track our results. Click anywhere on the canvas to open the widget box. All available widgets are located here. Now select the buttons.

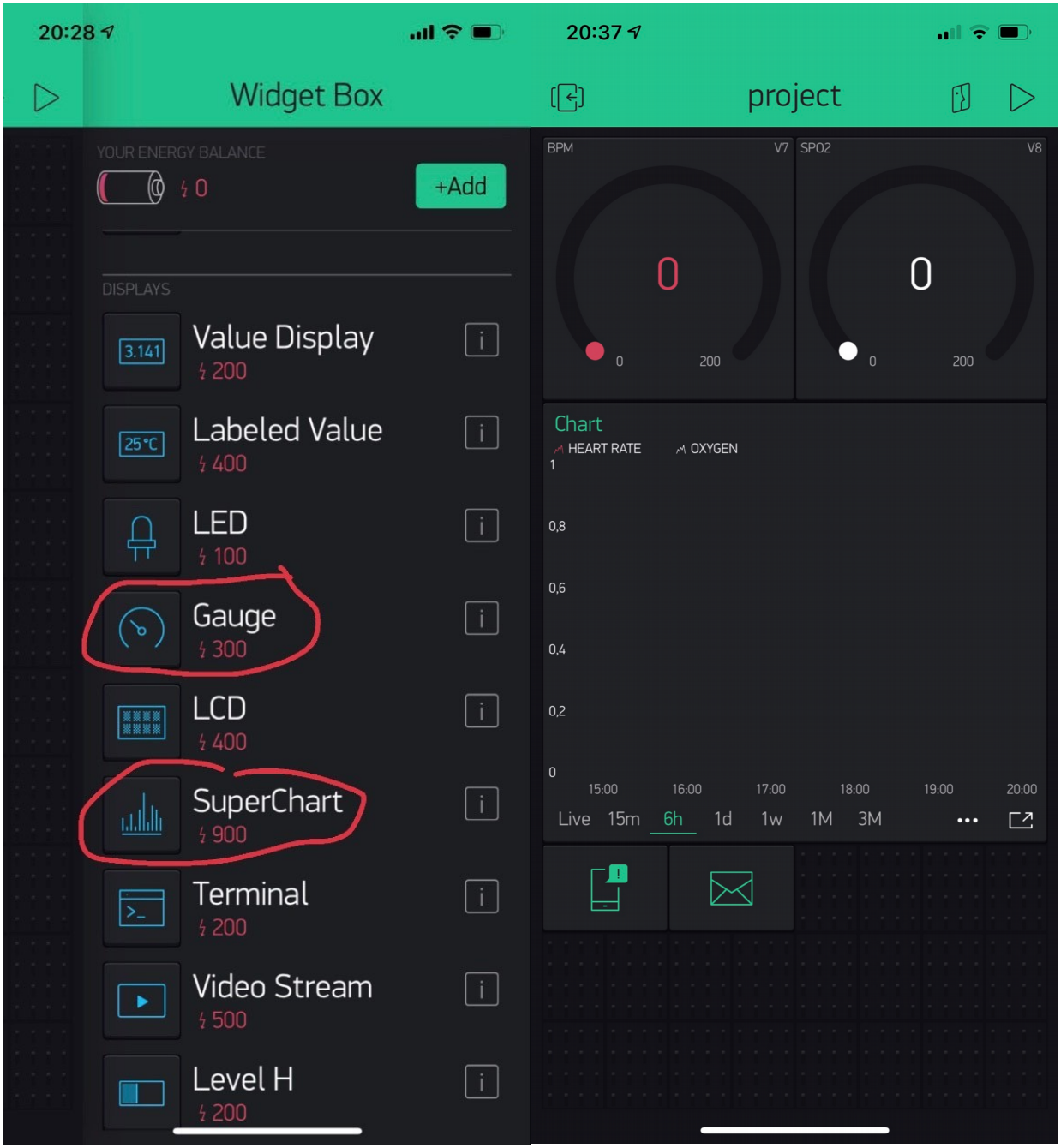


Figure 14: Widget Box

- **Drag-n-Drop**: Press and hold Widget to drag it to a new location.
- **Widget Settings**: Each Widget has its own setting. Click on widgets to access them.

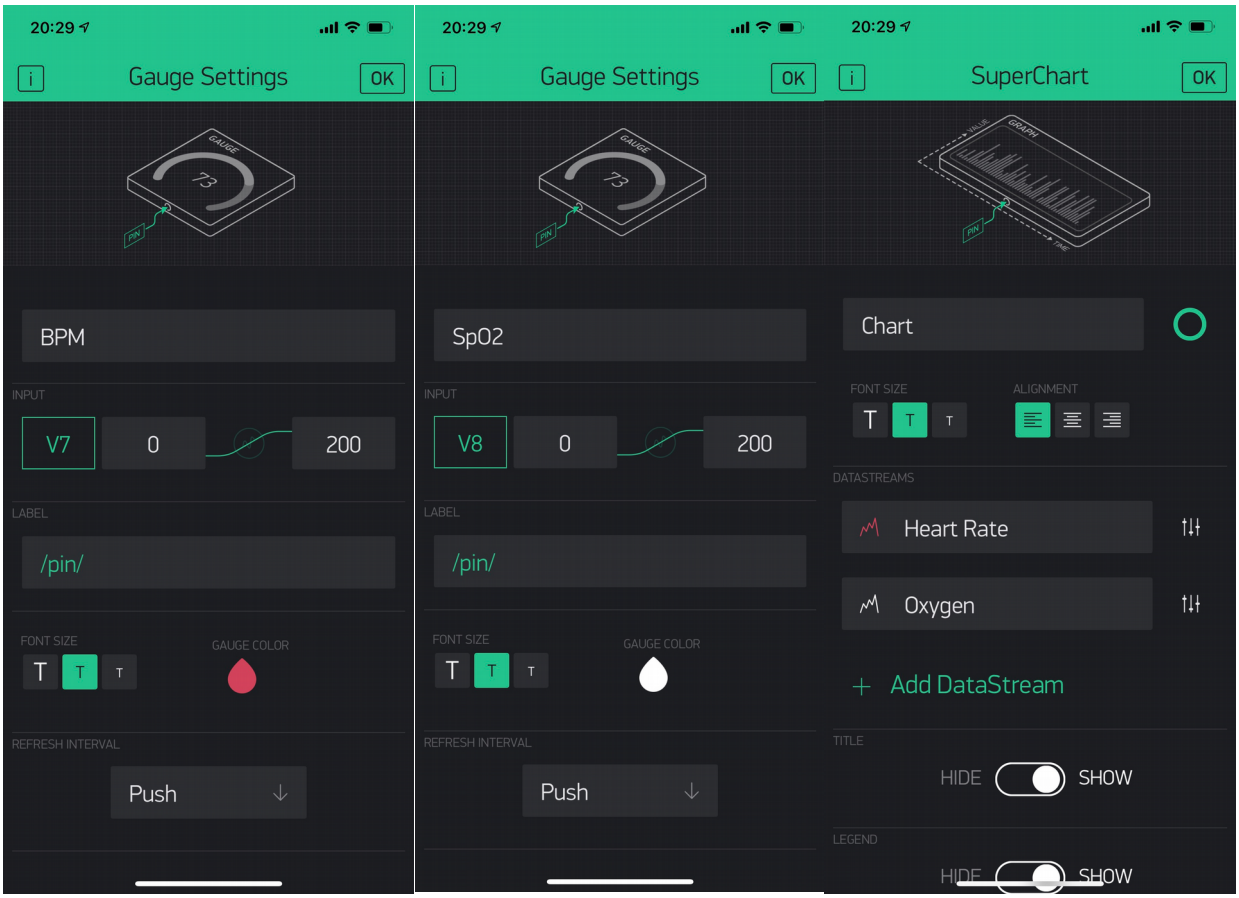


Figure 15: Notification setting

Step 6: Run project

When you are finished setting - press the PLAY button. This will switch you from EDIT mode to PLAY mode, where you can interact with the hardware. When in PLAY mode, you won't be able to drag or set up new widgets, press STOP and go back to EDIT mode.

1.4. Arduino IDE

- Connect to NodeMCU esp8266 board:
  - From the Arduino IDE, choose File -> Preferences in the menu bar
  - Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json in the Additional Board Manager URLs box.

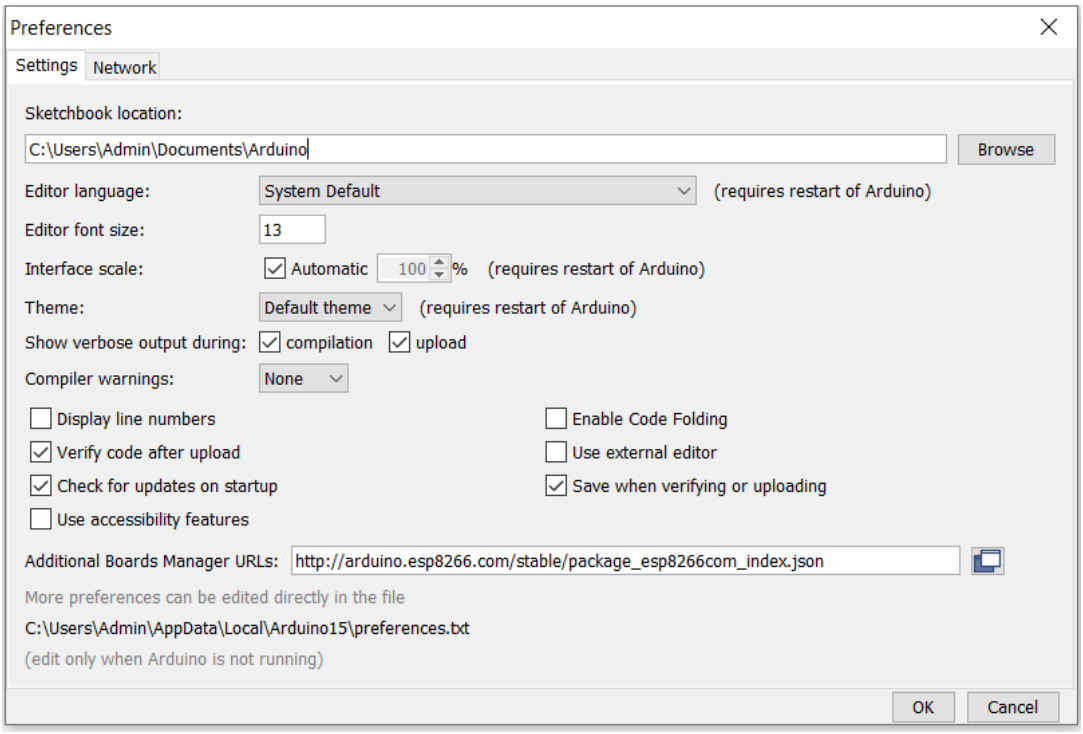


Figure 16: Preferences UI

- Select Tools -> Board menu -> Boards Manager and search for esp8266 platform.

Figure 17: Board Manager

- Select the version from Drop-down list.
- Click install.

And now, we can use NodeMCU by ESP8266 frameworks

- Import Library:

Select Tools -> Manage Libraries... -> Library Manager and search for correspond libraries

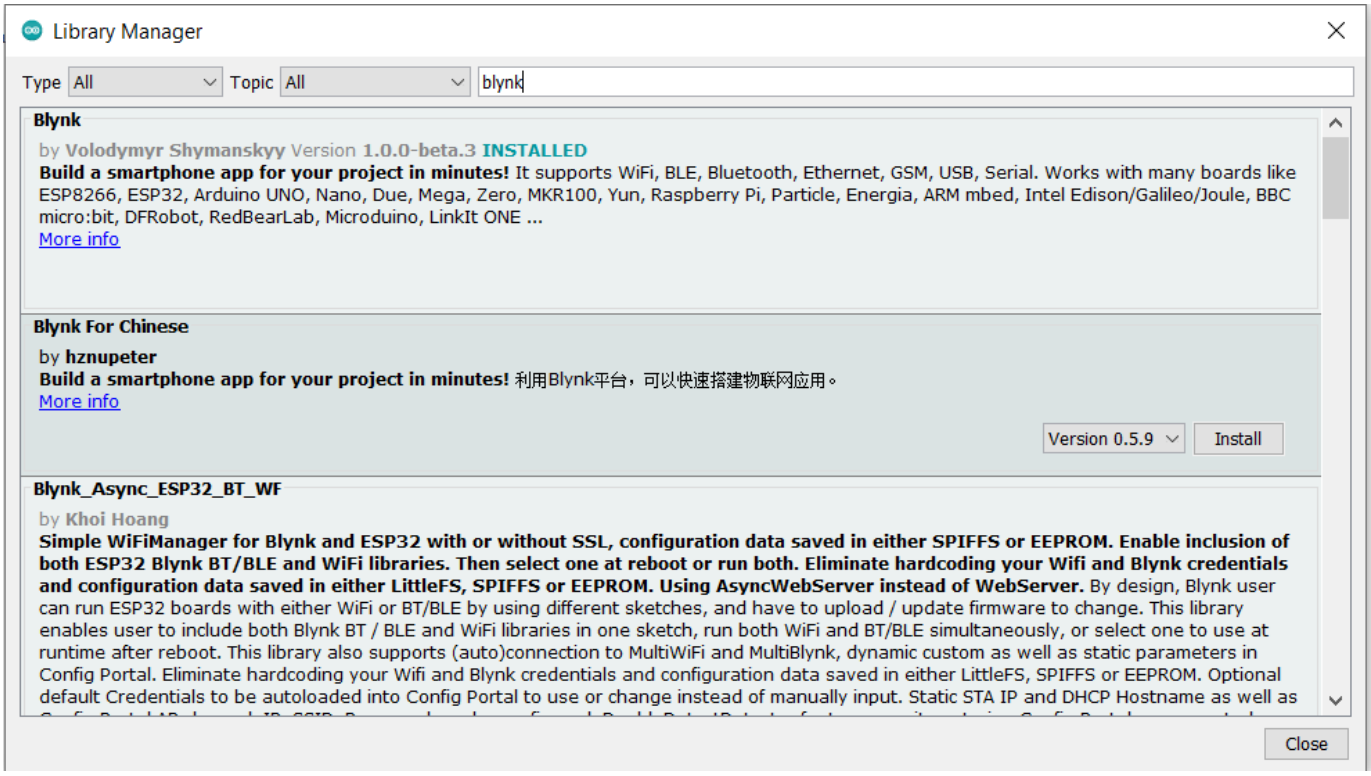


Figure 18: Library Manager

2. Design

2.1. Schematic

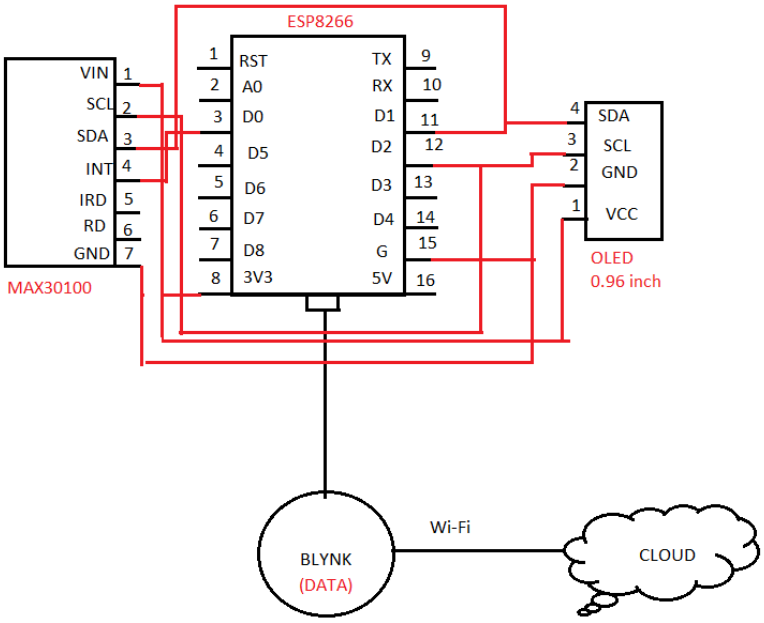


Figure 19: Schematic

NodeMCU ESP8266 pin	OLED 0.96 inch	MAX30100
3V	VCC	VIN
D0		INT
D1	SDA	SDA
D2	SCL	SCL
G	GND	GND

Table 1: Circuit Pin structure

2.2. Specification

Hardware specifications	
At mega Microcontroller	NodeMCU ESP8266
Cables and Connectors	USP Cables
PCB and Breadboards	Buzzer
Transformer/Adapter	Reset Button
Software Specifications	
Arduino Compiler	Programming Language: C
Data Process Platform	Blynk

Table 2: Project Specification

2.3. Sketch

To collect digital data about the accelerometer and gyroscope we'll need the “Wire.h” library, but it's one of the standard libraries from the Arduino framework so we just need to call it to use. - To connect Blynk application, we will use Blynk library on Arduino IDE. In the navigation bar we hover on “Tools” and select “Library Management...” and a library management panel is displayed. Look in the search bar and type "Blynk" and set the first element to the Blynk library.

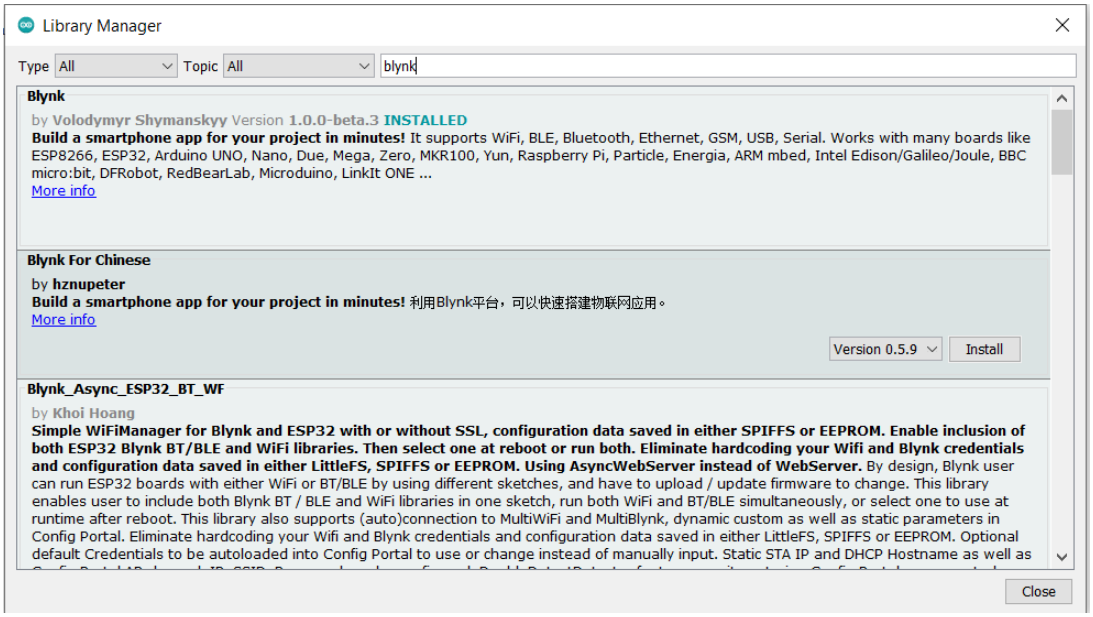


Figure 20: Blynk installation

Upload the run code to command our system. In particular, we have three interchangeable components for the Wi-Fi protocol.

- ssid: your wifi name
- pass: your password wifi
- auth: The token is taken from the Blynk application project

3. Code and implement

```
#include "MAX30100_PulseOximeter.h"
#define BLYNK_PRINT Serial
#include <Blynk.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "Wire.h"
#include "Adafruit_GFX.h"
#include "OakOLED.h"

#define REPORTING_PERIOD_MS 1000
OakOLED oled;

char auth[] = "TLStH5g19Xvs3Zxesqq2HMBS1DRSHP5k";            // You should get Auth Token in the Blynk App.
char ssid[] = "Tang4";                                      // Your WiFi credentials.
char pass[] = "tang4@123";

// Connections : SCL PIN - D1 , SDA PIN - D2 , INT PIN - D0
PulseOximeter pox;

float BPM, SpO2;
uint32_t tsLastReport = 0;

const unsigned char bitmap [] PROGMEM=
{
0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x18, 0x00, 0x0f, 0xe0, 0x7f, 0x00, 0x3f, 0xf9, 0xff, 0xc0,
0x7f, 0xf9, 0xff, 0xc0, 0x7f, 0xff, 0xff, 0xe0, 0x7f, 0xff, 0xff, 0xe0, 0xff, 0xff, 0xff, 0xf0,
0xff, 0xf7, 0xff, 0xf0, 0xff, 0xe7, 0xff, 0xf0, 0xff, 0xe7, 0xff, 0xf0, 0x7f, 0xdb, 0xff, 0xe0,
0x7f, 0x9b, 0xff, 0xe0, 0x00, 0x3b, 0xc0, 0x00, 0x3f, 0xf9, 0x9f, 0xc0, 0x3f, 0xfd, 0xbf, 0xc0,
0x1f, 0xfd, 0xbf, 0x80, 0x0f, 0xfd, 0x7f, 0x00, 0x07, 0xfe, 0x7e, 0x00, 0x03, 0xfe, 0xfc, 0x00,
0x01, 0xff, 0xf8, 0x00, 0x00, 0xff, 0xf0, 0x00, 0x00, 0x7f, 0xe0, 0x00, 0x00, 0x3f, 0xc0, 0x00,
0x00, 0x0f, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
```

```
void onBeatDetected()
{
    Serial.println("Beat Detected!");
    oled.drawBitmap( 60, 20, bitmap, 28, 28, 1);
    oled.display();
}

void setup()
{
    Serial.begin(115200);
    oled.begin();
    oled.clearDisplay();
    oled.setTextSize(1);
    oled.setTextColor(1);
    oled.setCursor(0, 0);

    oled.println("Initializing pulse oximeter..");
    oled.display();

    pinMode(16, OUTPUT);
    Blynk.begin(auth, ssid, pass);

    Serial.print("Initializing Pulse Oximeter..");
```

```
    if (!pox.begin())
    {
         Serial.println("FAILED");
         oled.clearDisplay();
         oled.setTextSize(1);
         oled.setTextColor(1);
         oled.setCursor(0, 0);
         oled.println("FAILED");
         oled.display();
         for(;;);
    }
    else
    {
         oled.clearDisplay();
         oled.setTextSize(1);
         oled.setTextColor(1);
         oled.setCursor(0, 0);
         oled.println("SUCCESS");
         oled.display();
         Serial.println("SUCCESS");
         pox.setOnBeatDetectedCallback(onBeatDetected);
    }

    // The default current for the IR LED is 50mA and it could be changed by uncommenting the following line.
     //pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

}
```

```
void loop()
{
    pox.update();
    Blynk.run();

    BPM = pox.getHeartRate();
    SpO2 = pox.getSpO2();
     if (millis() - tsLastReport > REPORTING_PERIOD_MS)
    {
        Serial.print("Heart rate:");
        Serial.print(BPM);
        Serial.print(" bpm / SpO2:");
        Serial.print(SpO2);
        Serial.println(" %");

        Blynk.virtualWrite(V7, BPM);
        Blynk.virtualWrite(V8, SpO2);

        oled.clearDisplay();
        oled.setTextSize(1);
        oled.setTextColor(1);
        oled.setCursor(0,16);
        oled.println(pox.getHeartRate());

        oled.setTextSize(1);
        oled.setTextColor(1);
        oled.setCursor(0, 0);
        oled.println("Heart BPM");

        oled.setTextSize(1);
        oled.setTextColor(1);
        oled.setCursor(0, 30);
        oled.println("Spo2");
```

```
            oled.setTextSize(1);
            oled.setTextColor(1);
            oled.setCursor(0,45);
            oled.println(pox.getSpO2());
            oled.display();

            tsLastReport = millis();
        }
    }
```

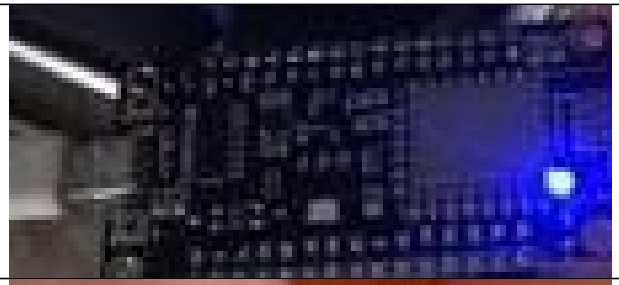
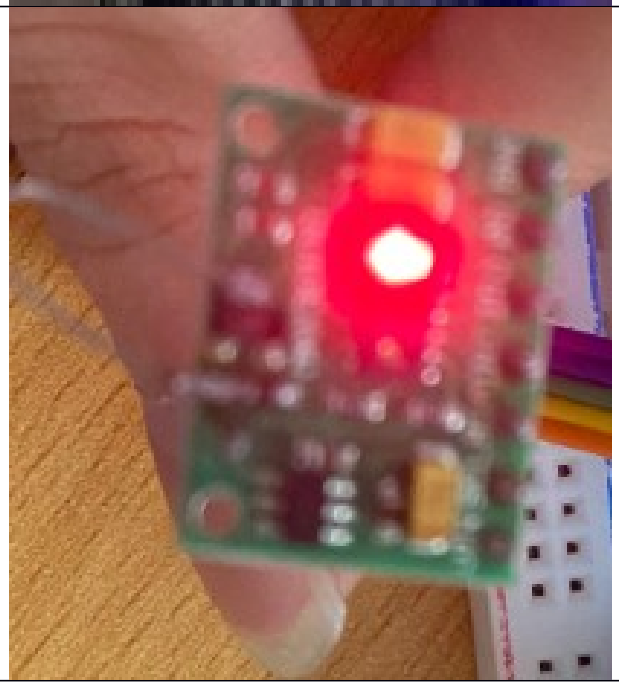
Figure 21: Code and Implement

4. Demo project

4.1. Testing

For specifics, we should set up a test case board to make sure each element in our product works as well as what we recommend for this project. I decided to split the testing phase into four phases:

- Unit test.
- Integration testing.
- System test.
- Checking devices.

Stage	ID	Test case	Expected result	Actual result	Status	Note
Unit Testing	1	NodeMCU working quality.	Receives input power by the onboard LED signal.		Done	
	2	MAX30100 sensor	Receives input power by the LED signal in the sensor.		Done	

	3	OLED	Screen will appear		Fail	
Integration testing	4	Connectivity capability of NodeMCU.	[1078] Connecting to Tang4 [5914] Connected to WiFi [5914] IP: 192.168.0.175	[1078] Connecting to Tang4 [5914] Connected to WiFi [5914] IP: 192.168.0.175	Done	Check it on Serial monitor in Arduino IDE.
	5	Blynk connectivity.	[5914] v1.0.0-beta.3 on NodeMCU [5921] Connecting to blynk-cloud.com:80 [7138] Ready (ping: 105ms). Initializing Pulse Oximeter..SUCCESS	[5914] v1.0.0-beta.3 on NodeMCU [5921] Connecting to blynk-cloud.com:80 [7138] Ready (ping: 105ms). Initializing Pulse Oximeter..SUCCESS	Done	

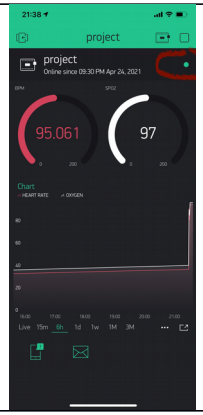
	6	Identify data about results determined by changing one point (as sensor position point)	Heart rate:10.36 bpm / SpO2:0.00 % Heart rate:10.36 bpm / SpO2:0.00 % Beat Detected! Heart rate:16.76 bpm / SpO2:0.00 % Heart rate:16.76 bpm / SpO2:0.00 % Beat Detected! Beat Detected! Heart rate:42.23 bpm / SpO2:96.00 % Beat Detected! Heart rate:49.55 bpm / SpO2:96.00 % Heart rate:49.55 bpm / SpO2:96.00 %	Heart rate:10.36 bpm / SpO2:0.00 % Heart rate:10.36 bpm / SpO2:0.00 % Beat Detected! Heart rate:16.76 bpm / SpO2:0.00 % Heart rate:16.76 bpm / SpO2:0.00 % Beat Detected! Beat Detected! Heart rate:42.23 bpm / SpO2:96.00 % Beat Detected! Heart rate:49.55 bpm / SpO2:96.00 % Heart rate:49.55 bpm / SpO2:96.00 %	Done	
System testing	7	The success of embedded sketch is in NodeMCU.	"Beat Detected!" when don't touch in sensor	Beat Detected!	Done	
	8	Blynk notification	Warning with green light after NodeMCU successfully connected.		Done	
Device testing	9	Testing the product.	The buzzer is active when Heart Rate detection is done	Initial sound calculator is 12db Buzzer sound calculator is 51db	Done	I used the audio sensor to calculate the sound from our device.

Figure 22: Test cases

5. Link Video:

https://youtu.be/Z_gWURy9zUc

III. Review & Feedback

1. Review

To collect user reviews, we have implemented a number of popular ways such as: multiple choice questions, a fall detector forum, etc. by creating and publishing several groups, forums and media. current society

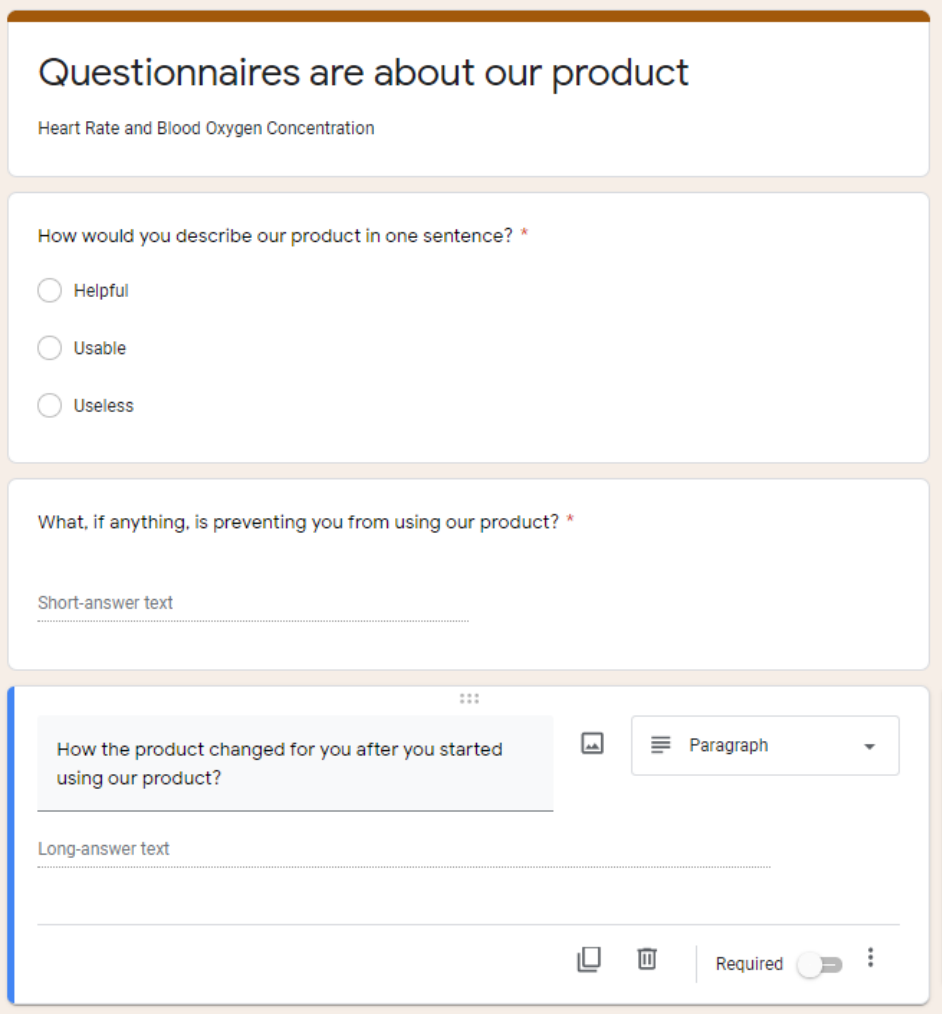


Figure 23: Multiple-choice question

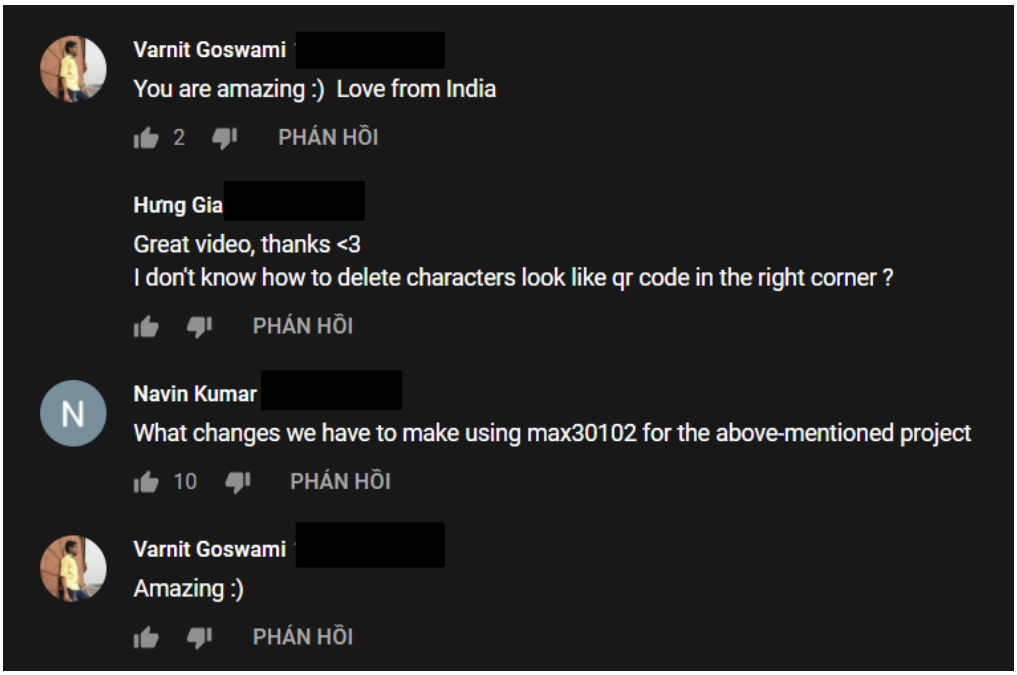


Figure 24: Project Detector Comments

After testing and demo our product to users and some survey, we've collected and clarified them in the following table:

Good point	Bad point
The product is very useful and usable.	The product was not detected as quickly as possible.
It helps people to focus more on their work without being overly concerned about the old.	They are in doubt about the low cost of this product.
Its price is cheaper than they thought	
There is no need to wait for the device to reset after it detects any abnormality	
It makes the elderly more comfortable while sleeping regardless of whether they have a heart attack during sleep like stroke	

Figure 25: Good and bad points

2. Evaluate end user feedback from your IoT application

2.1. Evaluate usability and commercialization

In general, we need to have a clear critical rating ratio using a given chart. The following chart is a general summary of the product's essentials in the current market.

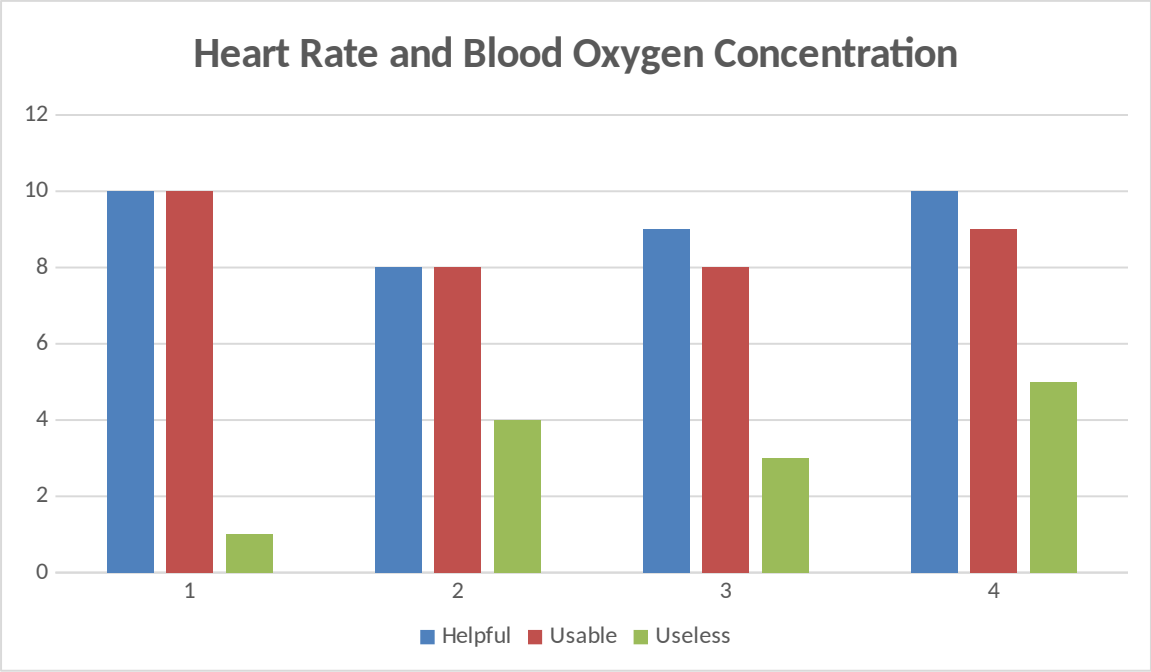


Figure 26:The column chart

Looking at the usability chart, we can see a positive increase in usability after each gathering and reviewing it. It has proven that we can face and get stuck in the current market in developing an IoT device such as the Heart Rate and Blood Oxygen Concentration device.

2.2. Feasibility evaluation

We mentioned the difficulties and challenges that this project caused when implementing the plan such as:

- **Technical**:
  - **Technical department**: has a basic knowledge base and is good at defining the structure of technology products and trying to catch up with current IoT technology trends. But the optimal utility and cost and the product are not found in the market

- **Application programmer**: has basic knowledge of creating application interfaces but lacks experience with functions of IoT devices.
- **Project scope**: an underlying product has an underlying architecture, so its risk is moderate
- **Compatibility**: this is the first version of this product if the identification of the components of this device is successful, we have completed the inception era of this project.

- **Economic**:
  - Operating cost: calculated with hardware and software cost investment for productivity is over 1,000,000 VND.
  - Trade cost: selling cost is 250,000 VND.
  - This project is created to support, help and support those who have difficult circumstances and have good conditions to study.
  - Intangible costs and benefits: the team's first milestone is to create the first IoT device and to apply it to the real-time world.
- **Organization:**
  - This project is implemented in collaboration with interested partners with a real-time project.
  - The branch of the organization is only in its establishment period, so we decided to improve this product in a more powerful and optimal way to announce our reputation in this market.

2.3. Ability to improve

- In the next versions, we will gradually reach out to the customer and add the missing functionality on the Blynk platform.
- In the next version (3.0.0), we will add a standalone application platform to control the hardware instead of based on the application Blynk is being applied.

3. Development plan

Plan title	Heart Rate and Blood Oxygen Concentration ver. 2	
Goal	Building applications for measuring the heart rate and the amount of oxygen in the blood	
Milestone Strategy	Planning -> Analysis -> Design -> Implement	
Measure	Performance + 10% of agreed cost/schedule/feature milestones	
Team department	Management and Analysis	Luu Vinh Truc Phuong
		Tran Trong Minh
	Design	Mai Thanh Tien Dung
		Nguyen Tran Hoang Nhi
	Programming	Nguyen Van Thang
		Hoang Xuan Loc

Phase	Action	Department	Duration (days)	Resources and Methods	Status
Planning	Initial system requirements	Management and Analysis	3	Identify opportunities	Opening
	Feasibility study	Management and Analysis	3	- Technical feasibility - Possible economy - Organizational feasibility	Opening
	Project plan	Management and Analysis	6	- Estimated time - Mission identification - Work division structure - PERT chart - Gantt charts - Management scope - Project staff - Risk management	Opening
Analysis	Definition required	Management and Analysis	10	- Interview - JAD session - Table of questions - Observe	Opening
	Use cases	Management and Analysis	Predicting	Use case analysis	Closing
	Process Models	Management and Analysis Programming	Predicting	Mapping the data flow	Closing
	Data Model	Management and Analysis	Predicting	Entity relationship model	Closing

		Programming		Normalize	
Design	Substitution Matrix	Design	Predicting	Design strategy	Closing
	Architectural design	Design	Predicting	Architectural design	Closing
	Design interface	Programming	Predicting	- Use script - Interface structure - Interface standards - Interface prototype - Interface evaluation	Closing
	Physical process model	Design	Predicting	Mapping the data flow	Closing
	Hardware/software specification	Design	Predicting	Select hardware and software	Closing
	Physical data model	Design	Predicting	- Entity relationship model - Standardized - Performance tuning - Estimated size	Closing
	Design program	Programming	Predicting	- Program structure chart - Program specification	Closing
	Database and file specification	Design	Predicting	Select the data format	Closing
Implementation	Test plan	Management and Analysis	Predicting	Software Testing	Closing
	Move plan	Management and Analysis	Predicting	Choose a conversion strategy	Closing
	Support plan	Management and Analysis	Predicting	Selection assistance	Closing

Table 3: Development Plan

IV. References

AVAILABLE, E. K., 2014. *Pulse Oximeter and Heart-Rate Sensor IC*, s.l.: Maxim Integrated Products.

N.T., T., 2020. *IoT2*, Danang: FGW.

INDEX OF COMMENTS