

## รายงาน circuit project

### 1.รูปเล่ม

1.ภาพรวมโครงงาน

2.การวิเคราะห์วงจร

3.การเชื่อมต่อกับวงจรไมโครคอนโทรลเลอร์

4.ผลการทดสอบ

5.โปสเตอร์

6. Source code

### 2.Poster

### 3.ชิ้นงาน

# 1. ภาพรวมโครงการ

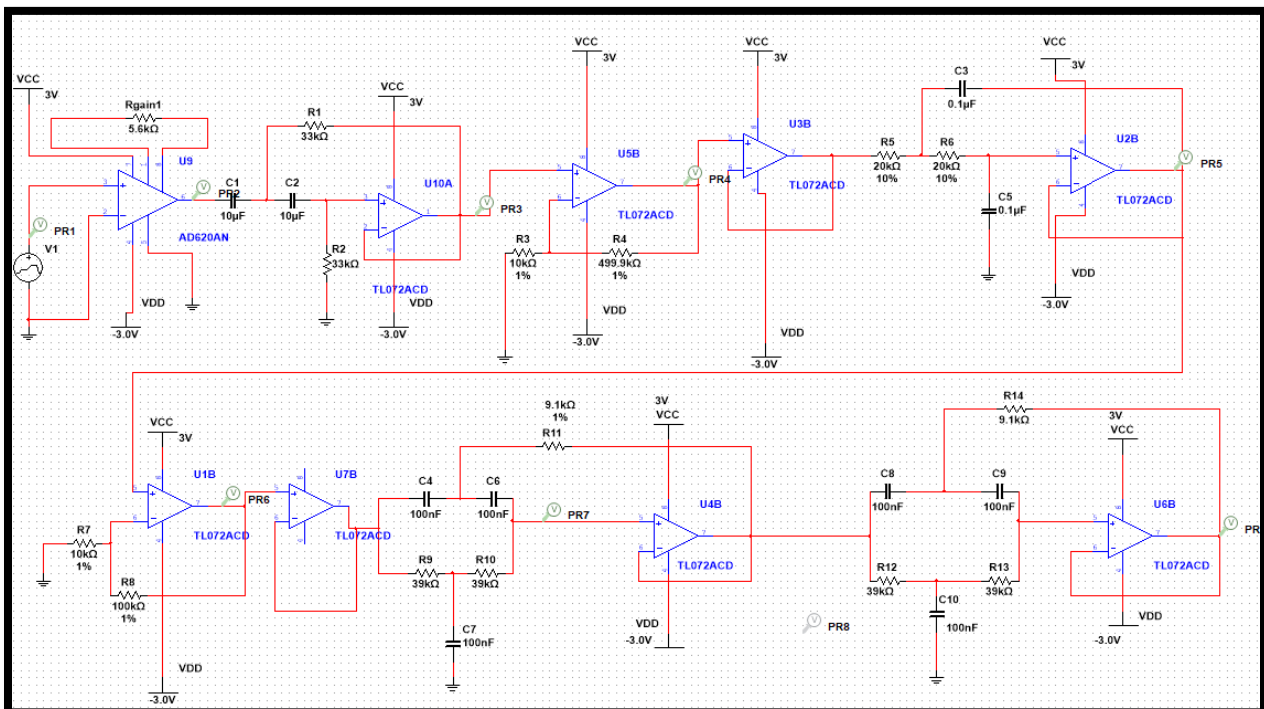
## 1.1 ที่มาและความสำคัญ

กลุ่มของเราเล็งเห็นถึงความสำคัญของปัญหาโรคหัวใจ สถิติสาธารณสุขของประเทศไทยในปี 2564 พบว่า "โรคหัวใจขาดเลือด" เป็นสาเหตุการเสียชีวิตของคนไทยมากเป็นอันดับ 4 รองจากโรคมะเร็ง โรคหลอดเลือดในสมอง และปอดบวม โดยโรคหัวใจขาดเลือด คร่าชีวิตคนไทยปีละประมาณ 20,000 คน หรือราว 33 คน ต่อประชากร 1 แสนคน

นอกจากนี้ในช่วง 16 ปี วิทยาลัยการแพทย์อเมริกัน ได้ทำการศึกษา (2000- 2016) สัดส่วนของคนอายุน้อยที่หัวใจวาย เพิ่มขึ้น 2% ต่อปีในช่วง 10 ปีหลัง

นี่คือแรงบันดาลใจของพวกเราในการจะสร้างนวัตกรรมเกี่ยวกับคลื่นหัวใจ ซึ่งมันน่าจะดีไม่น้อยทีเดียวหากว่าเรานั้นสามารถเช็คคลื่นหัวใจของตนเองนั้นได้ตลอดเวลา และที่สำคัญคือเราสามารถเช็คดูได้ง่ายๆผ่านโทรศัพท์มือถือ นั่นเป็นเหตุผลที่เพียงพอแล้วที่เราจะสร้างนวัตกรรมที่มีชื่อว่า **Heart Rate, HRV And ECG Monitoring Via Blynk App**

## 1.2 วงจรสำหรับโปรเจค

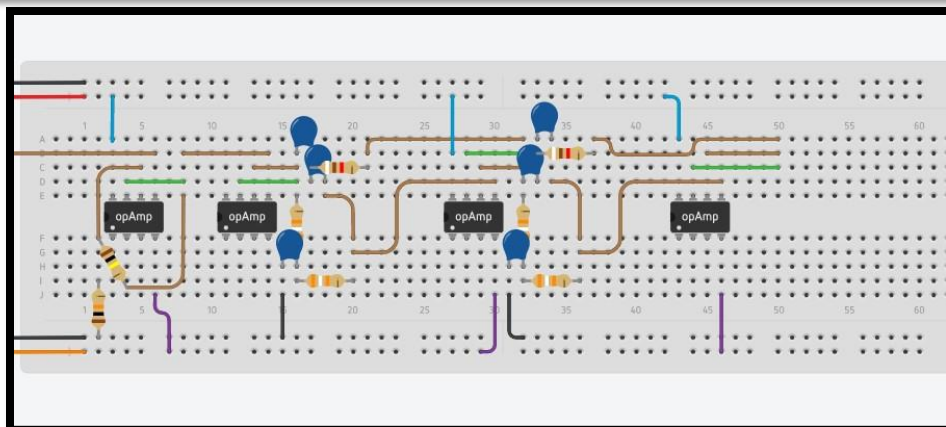
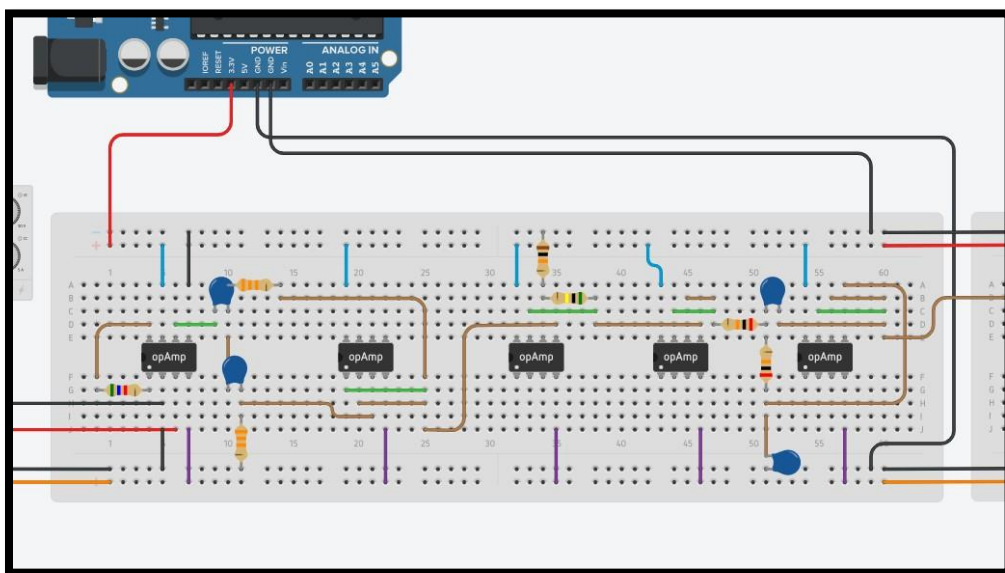
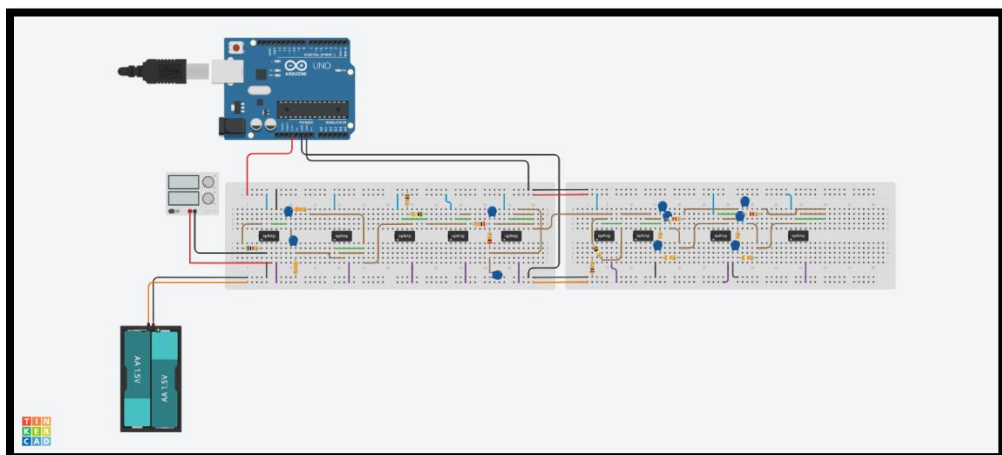


### 1.3.อุปกรณ์

|                             |       |
|-----------------------------|-------|
| - บอร์ด ESP 8266            | 1 ตัว |
| - เซนเซอร์ Electrode Pad    | 1 ตัว |
| - AD620                     | 1 ตัว |
| - TL072ACD                  | 8 ตัว |
| - 5.6 k $\Omega$ Resistor   | 1 ตัว |
| - 33 k $\Omega$ Resistor    | 1 ตัว |
| - 10 k $\Omega$ Resistor    | 1 ตัว |
| - 499.9 k $\Omega$ Resistor | 1 ตัว |
| - 20 k $\Omega$ Resistor    | 2 ตัว |
| - 100 k $\Omega$ Resistor   | 1 ตัว |
| - 39 k $\Omega$ Resistor    | 4 ตัว |
| - 9.1 k $\Omega$ Resistor   | 2 ตัว |
| - 10 $\mu$ F Capacitor      | 2 ตัว |
| - 0.1 $\mu$ F Capacitor     | 2 ตัว |
| - 100 nF Capacitor          | 6 ตัว |

#### 1.4. การออกแบบวงจร

การออกแบบวงจรนั้น จะออกแบบในเว็บที่มีชื่อว่า **Tinkercard** เพื่อช่วยให้วงจรก่อนจะต่อจริงนั้น มีความเป็นระเบียบเรียบร้อย และลดข้อผิดพลาดได้ดียิ่งขึ้นอีกด้วย โดยวงจรที่ได้ออกแบบไว้มีรูปแบบดังรูป



## 1.5 ขั้นตอนการทำงานของระบบ

1.5.1 Input : โดยส่วนของ Input นั้น จะรับค่าจากเซนเซอร์ที่มีชื่อ Electrode Pad ซึ่งค่าที่ได้จะเป็นกราฟ

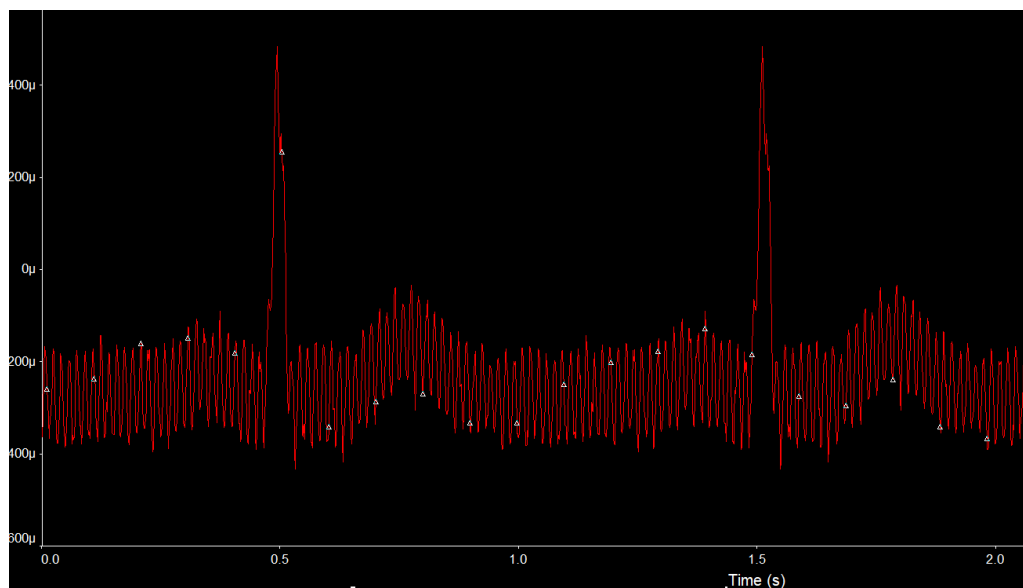
คลื่นไฟฟ้า (ECG) นั้นเอง



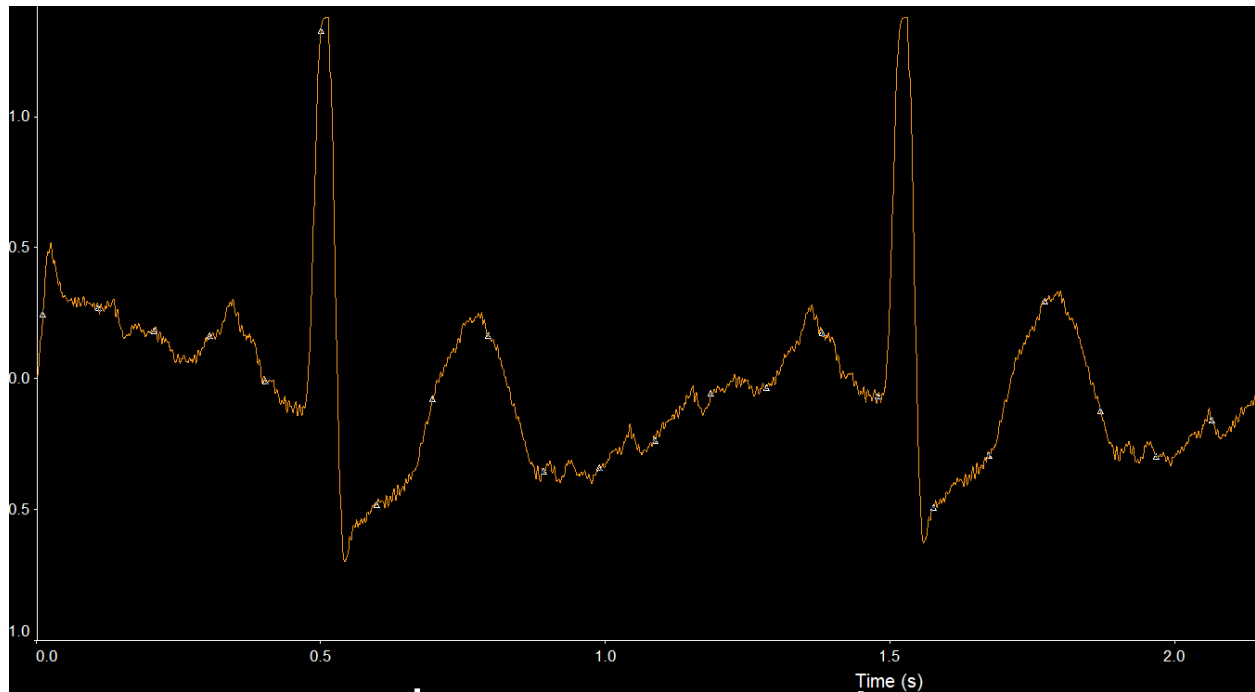
1.5.2 Process : จะนำกราฟ (ECG) ที่ได้นั้นมาผ่านวงจรที่ได้ออกแบบไว้ ซึ่งในวงจรนั้นจะเป็นวงจร Op-amp ซึ่ง

ทำหน้าที่ขยายรูปกราฟและกรองความถี่ ทำให้เรานั้นสามารถนำข้อมูลไปใช้งานและตรวจสอบข้อมูลต่างๆได้ง่ายขึ้นและมีความแม่นยำอีกด้วย

- กราฟที่ได้รับจาก ECG

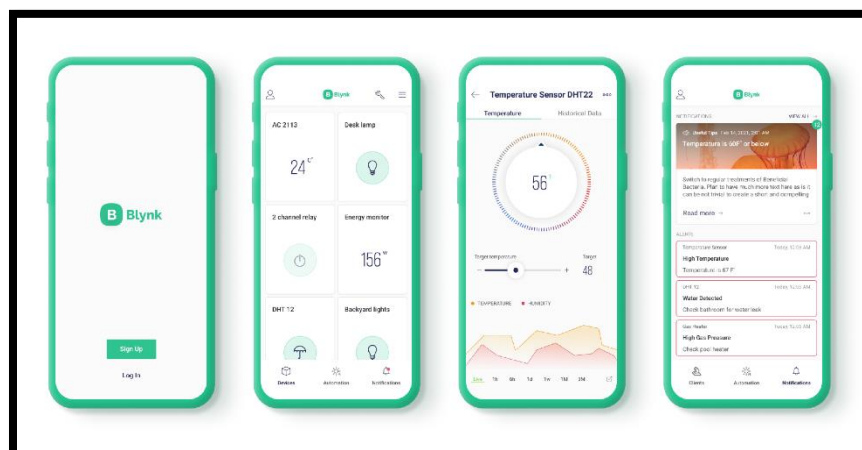


- กราฟที่ผ่านวงจรที่เราได้ออกแบบไว้



1.5.3 Output : หลังจากเราได้รูปกราฟที่ผ่านวงจรที่เราออกแบบไว้แล้วนั้น เราจะนำข้อมูลต่างๆ Plot ผ่านบอร์ด

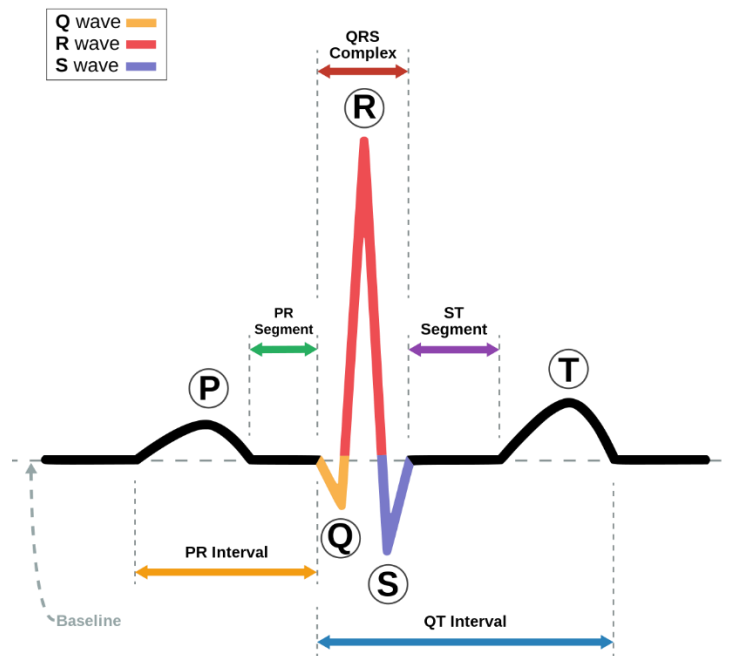
ESP 8266 และนำมาเชื่อมต่อกับ Wi-Fi ซึ่งจะทำให้เรานั้นสามารถตรวจค่า ECG ได้ตลอดเวลา  
ผ่านแอปที่มีชื่อว่า Blynk App ได้ตลอดเวลา



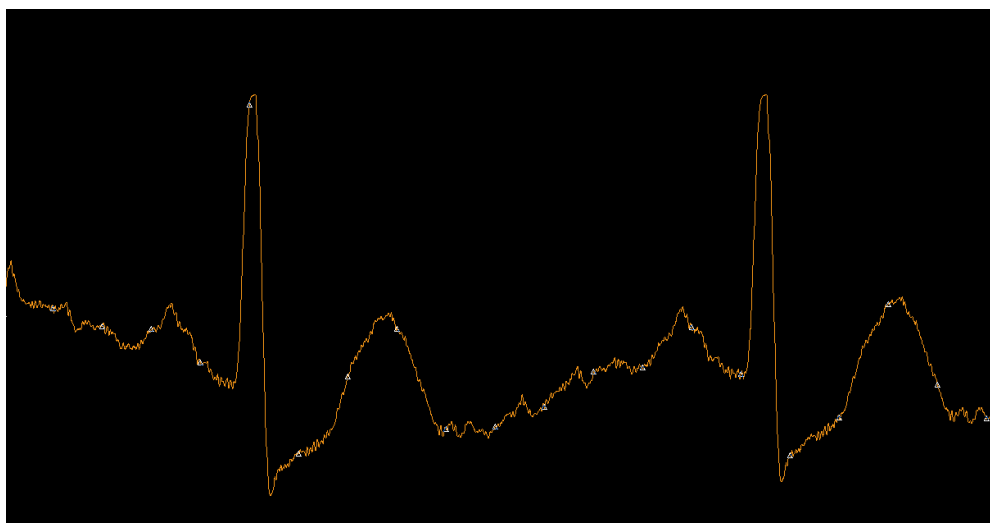
## 1.6 ประโยชน์และผลลัพธ์ที่จะได้

ประโยชน์ที่จะได้นั้น เราจะสามารถตรวจสอบคลื่นไฟฟ้าหัวใจ (ECG) ได้ตลอดทุกที่ทุกเวลาตามที่ต้องการ ซึ่งจะลดเวลาการเดินทางไปโรงพยาบาลรวมทั้งค่าใช้จ่ายต่างๆได้อีกด้วย แต่ถึงอย่างนั้นเราควรมีความรู้เกี่ยวกับการอ่านกราฟ ECG ด้วยเพื่อให้ผลลัพธ์นั้นตรงตามความต้องการของผู้ใช้งานให้มีประสิทธิภาพสูงสุดด้วยเช่นกัน

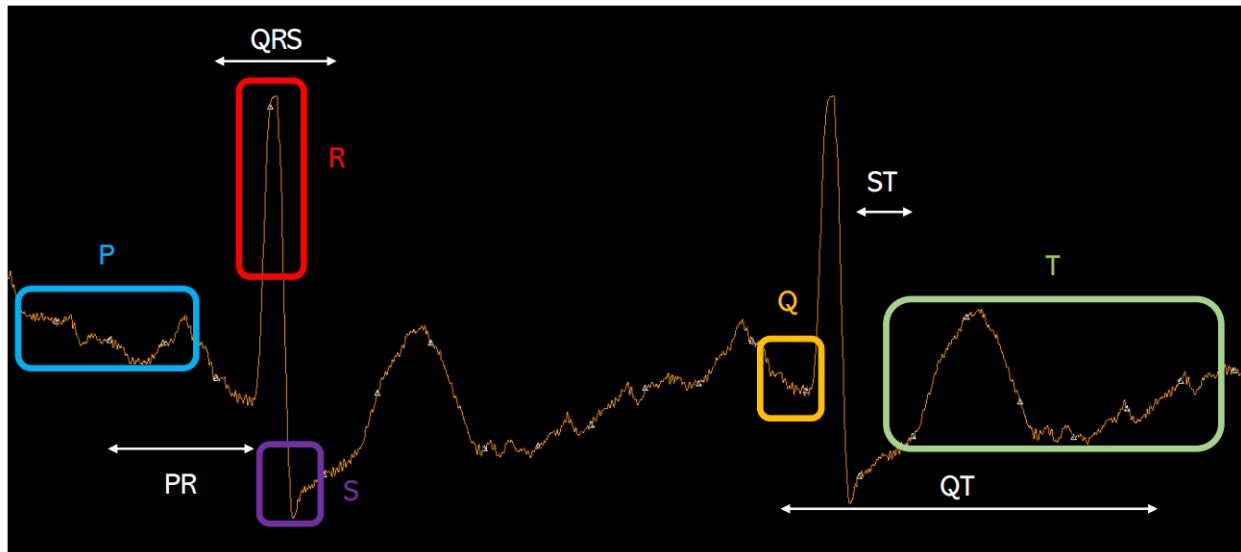
- กราฟ ECG



- ECG ของคนโดยทั่วไป (เมื่อผ่านวงจรที่เราออกแบบไว้แล้ว)



- ECG ของวงจร กับ ตำแหน่งกราฟและข้อมูลต่างๆ



- ECG กับการวินิจฉัยโรคต่างๆ เบื้องต้น

| รูปภาพ   | การวินิจฉัยเบื้องต้น  |
|--|---|
| ช่วงเวลา QT สั้น   | แคลเซียมสูงในเลือดเนื่องจากยาบางชนิด, ความผิดปกติทางพันธุกรรมบางอย่าง, ภาวะโพแทสเซียมสูง                        |
| ช่วงเวลา QT ยาว  | แคลเซียมสูงในเลือด, ยาบางชนิด, ความผิดปกติทางพันธุกรรมบางอย่าง  |
| คลื่น T แบนหรือคว่ำ  | หัวใจขาดเลือด, ภาวะโพแทสเซียมสูง, หัวใจห้องล่างซ้ายโตเกิน, ผลกระทบจากยาพวก <u>ดิจอกซิน</u> (Digoxin), ยาบางชนิด |
| คลื่น T เฉียบพลันสุดขีด  | อาจเป็นอาการแรกของกล้ามเนื้อหัวใจตายเฉียบพลัน, เมื่อคลื่น T กลายเป็นที่โดดเด่นมากขึ้น, สมมาตร, และแหลม          |
| คลื่น T ขึ้นสูงสุด, คลื่น QRS กว้าง, คลื่น PR ยาว, คลื่น QT สั้น | ภาวะโพแทสเซียมสูง, รักษาด้วย calcium chloride, กลูโคสและอินซูลิน หรือการล้างไต                                  |
| คลื่น U โดดเด่น  | ภาวะโพแทสเซียมสูง   |

จากข้อมูลข้างต้นทำให้เราเห็นถึงประโยชน์ของกราฟ คลื่นหัวใจไฟฟ้ามากขึ้นทำให้นั้นสามารถนำประโยชน์มาประยุกต์ใช้กับคนทั่วไปที่มีความสนใจได้อีกด้วย



## 2.การวิเคราะห์วงจร

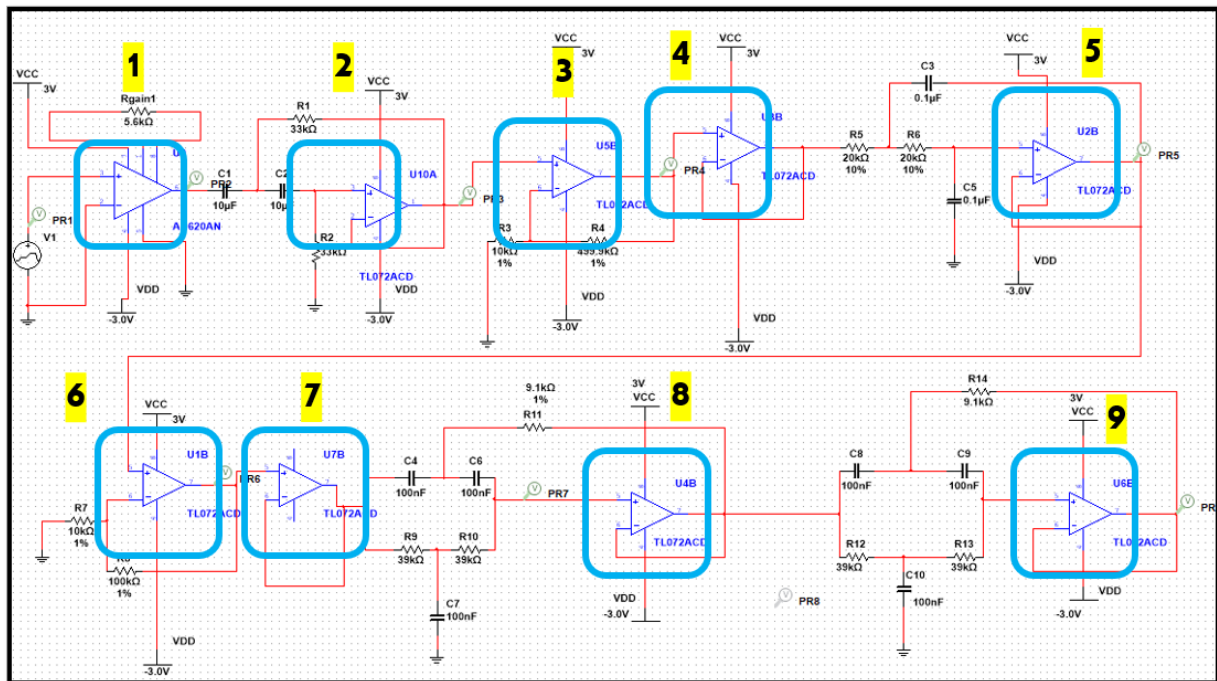
เนื่องจากวงจรของเรานั้นใช้ Op-amp จำนวน 9 ตัว เพื่อกรองความถี่รวมทั้งเพิ่มอัตราขยายกราฟให้ดูง่ายขึ้นโดยใช้ Op-amp จำนวน 2 ชนิด แบ่งเป็น

- AD620 1 ตัว (ตัวที่ 1)

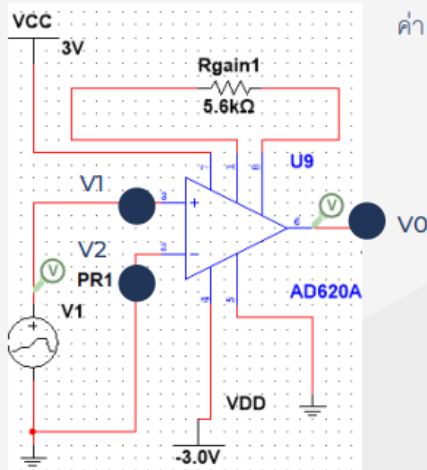
Function : Low Drift, Low Power Instrumentation Amp with Set Gains of 1 to 10000

- TL072 8 ตัว (ตัวที่ 2 ถึง 9)

Function : Dual Low-Noise JFET-Input Operational Amplifier



## 1. CMRR



ค่า CMRR ของ AD620A อยู่ในช่วง 100 - 120 dB

$$\text{CMRR} = 120 \text{ dB}$$

$$\text{CMRR} = 20 \log(\text{Ad}/\text{Ac})$$

$$120 = 20 \log(\text{Ad}/\text{Ac})$$

$$10^6 = (\text{Ad}/\text{Ac})$$

$$\text{Ac} = 10^{-6} \text{ Ad}$$

\*\* แต่  $\text{Ac} \ll \text{Ad}$

$\text{V0}$  ประมาณได้ว่า  $\text{AdV1}$

Common Mode

$$\text{Vic} = (\text{V1} + \text{V2})/2$$

$$\text{Vic} = \text{V1}/2$$

$$\text{Ac} = \text{Voc}/\text{Vic} = \text{Voc}/(\text{V1}/2)$$

$$\text{Ac} = 2(\text{Voc}/\text{V1})$$

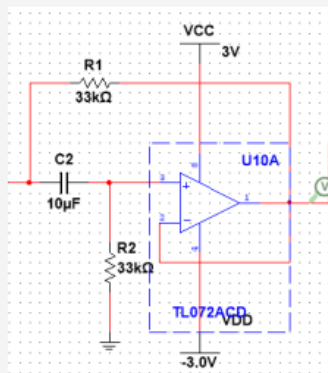
Differential Mode

$$\text{Ad} = \text{Vod}/\text{Vid} = \text{Vod}/(\text{V1} - \text{V2})$$

$$\text{Ad} = \text{Vod}/\text{V1}$$

$$\text{V0} = \text{AdV1} + ((\text{AcV1})/2)$$

## 2. HIGH - PASSFILTER



การคำนวณความถี่ Cutoff ในวงจร High-pass filter

$$\text{จาก } f_c = 1/(2(\pi)(R1 \cdot R2 \cdot C1 \cdot C2)^{1/2})$$

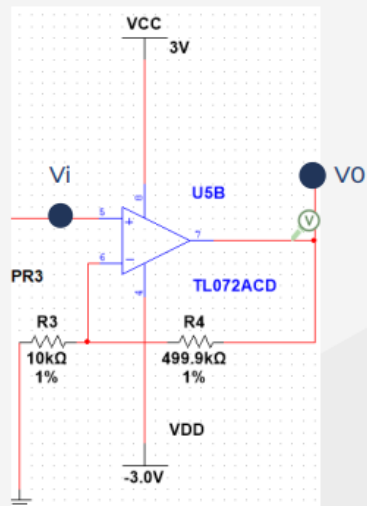
$$f_c = 1/(2(3.14)((33 \text{ k ohm})^2 \cdot (10 \mu\text{F})^2))^{1/2}$$

$$f_c = 1/(2 \cdot 3.14 \cdot 33 \text{ k ohm} \cdot 10 \mu\text{F})$$

$$f_c = 0.482 \text{ Hz}$$

\*\* ความถี่ที่มีค่าต่ำกว่า 0.482Hz จะผ่านไม่ได้

### 3. NON - INVERTING



การคำนวณหา Voltage gain ในวงจร non-inverting

$$V0/Vi = 1 + R4/R3$$

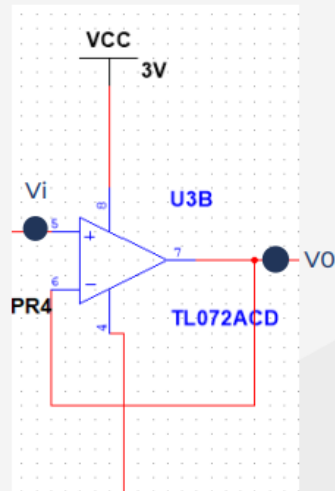
$$V0/Vi = 1 + (499.9 \text{ k ohm} / 10 \text{ k ohm})$$

$$V0/Vi = 1 + 49.99$$

$$Av = 50.99$$

\*\* Av มีค่าประมาณ 51 เท่า

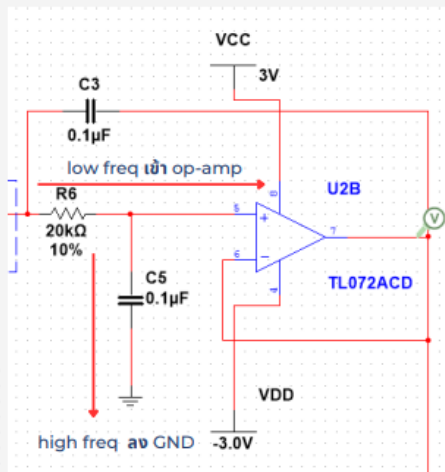
### 4. BUFFER



วงจร Buffer

\*\* ใช้ส่งผ่านแรงดัน Input ไปยัง Output

## 5. LOW - PASS FILTER



การคำนวณความถี่ Cutoff ในวงจร Low-pass filter

วงจรกรองสัญญาณความถี่ต่ำ

จาก  $f_{cutoff}$

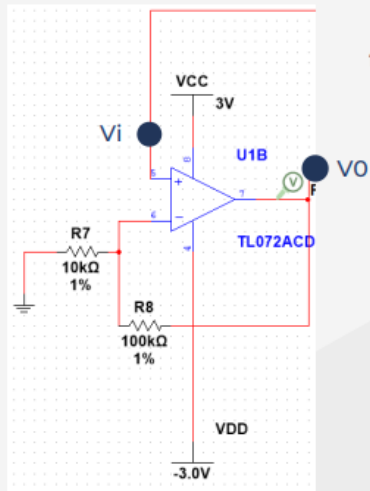
$$f_c = 1/(2(\pi)(R5 \cdot R6 \cdot C5 \cdot C6)^{1/2})$$

$$f_c = 1/(2(3.14)((20k \text{ ohm})^2 \cdot (0.1\mu F^2))^{1/2})$$

$$f_c = 79.58 \text{ Hz}$$

\*\* ความถี่ที่มีค่าสูงกว่า 79.58Hz จะผ่านไม่ได้

## 6. NON - INVERTING



การคำนวณหา Voltage gain ในวงจร non-inverting

$$V0/Vi = 1 + R8/R7$$

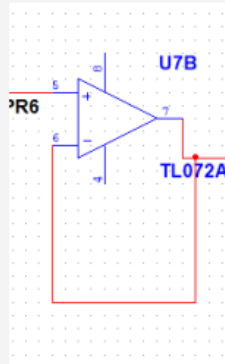
$$V0/Vi = 1 + (100k \text{ ohm}/10k \text{ ohm})$$

$$V0/Vi = 1 + 10$$

$$Av = 11$$

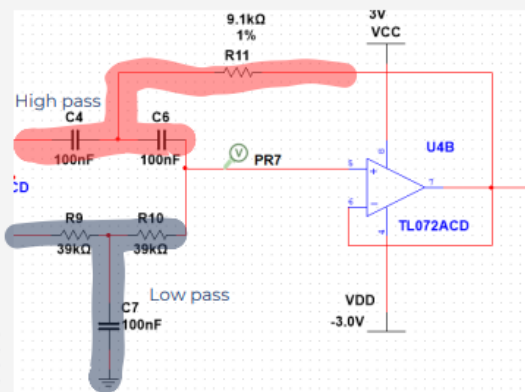
\*\*  $Av$  มีค่าประมาณ 11 เท่า

## 7 . BAND PASS FILTER



\*\* กรองความถี่ให้อยู่ในช่วงที่กำหนด

## 8 - 9 . NOTCH FILTER



การคำนวณหาความถี่ Cutoff และความถี่ notch  
ในวงจร notch filter

$$f_L = 1/(2(\pi)(R_9+R_{10})C_7)$$

$$f_L = 20.4 \text{ Hz}$$

$$f_H = 1/(2(\pi)(R_{11})C)$$

$$f_H = 349.8 \text{ Hz}$$

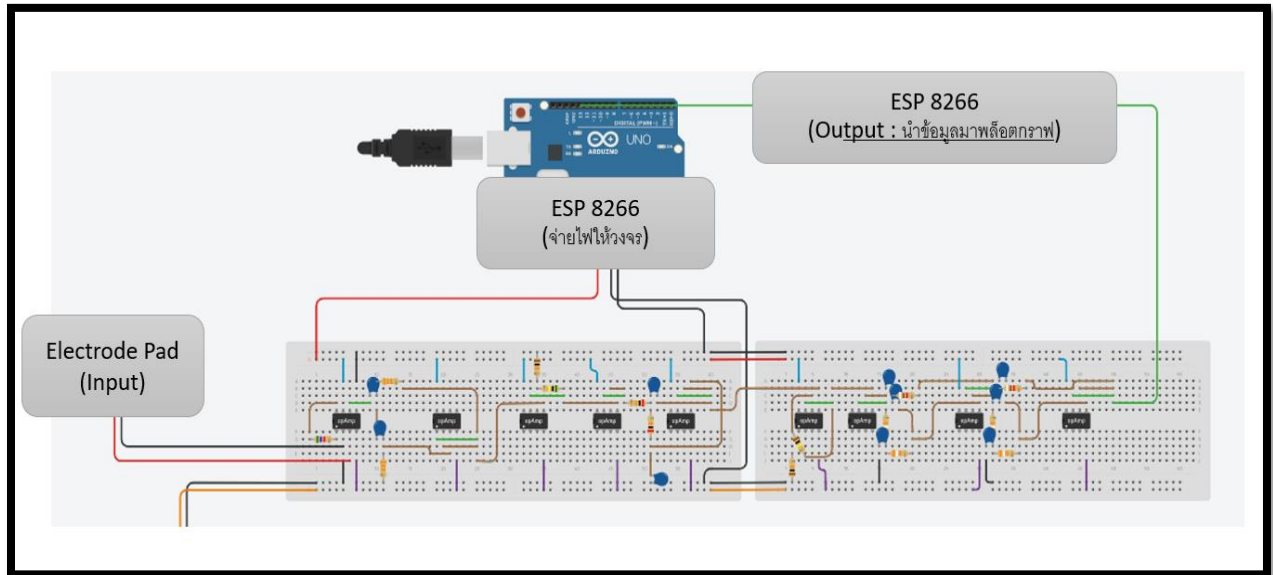
$$f_N = ((20.4)(349.8))^{1/2}$$

$$f_N = 84.5 \text{ Hz}$$

\*\* f notch = 84.5 Hz  
ความถี่ในช่วง 84.5 Hz  
จะผ่านไปยัง op-amp  
ไม่ได้

### 3.การเชื่อมต่อกับวงจรไมโครคอนโทรลเลอร์

#### 3.1 การเชื่อมต่อกับวงจรที่ออกแบบไว้



#### 3.2 การเชื่อมต่อกับวงจรจริง

#### 4.ผลการทดสอบ

## พบกับนวัตกรรม ECG VIA BLYNK APP

### จุดเด่น

- ไม่ว่าจะอยู่ที่ไหนก็สามารถตรวจสอบได้
- เช็กได้ทุกที่ทุกเวลาผ่านโทรศัพท์มือถือ
- รู้ทันโรคต่างๆผ่านคลื่นหัวใจไฟฟ้า (ECG)

ณ วันที่ 1 พฤษภาคม 2023





## 6.Source Code

### 6.1 Arduino

```
1 #include <ESP8266WiFi.h>
2 #include "PeakDetection.h"
3
4 const int LO_PLUS_PIN = D7; // AD8232 LO+ pin
5 const int LO_MINUS_PIN = D8; // AD8232 LO- pin
6 const int ECG_OUTPUT_PIN = A0; // AD8232 analog output pin
7
8 const int READ_RATE = 5; // Time (ms) between ECG readings
9 unsigned long lastReadTime = 0;
10
11 const int ROLLING_BUFFER_SIZE = 30; // Buffer size for calculating HRV
12 int rollingBuffer[ROLLING_BUFFER_SIZE];
13 int rollingBufferIndex = 0;
14
15 unsigned long lastPeakTime = 0;
16 const int MAXIMUM_TIME_BETWEEN_PEAKS = 2000; // Maximum time (ms) between peaks
17
18 PeakDetection peakDetector;
19
20 void setup() {
21     Serial.begin(115200);
22
23     pinMode(LO_PLUS_PIN, INPUT);
24     pinMode(LO_MINUS_PIN, INPUT);
25
26     for (int i = 0; i < ROLLING_BUFFER_SIZE; i++) {
27         rollingBuffer[i] = 0;
28     }
29
30     peakDetector.begin();
31 }
32
33 void loop() {
34     if (millis() - lastReadTime >= READ_RATE) {
35         int ecgValue = analogRead(ECG_OUTPUT_PIN);
36         Serial.print("ECG: ");
37         Serial.println(ecgValue);
38
39         if (digitalRead(LO_PLUS_PIN) == LOW || digitalRead(LO_MINUS_PIN) == LOW) {
40             Serial.println("Electrodes disconnected.");
41         }
42     }
43 }
```

```

39     if (digitalRead(LO_PLUS_PIN) == LOW || digitalRead(LO_MINUS_PIN) == LOW) {
40         Serial.println("Electrodes disconnected.");
41     }
42
43     peakDetector.add(ecgValue);
44
45     int heartRate = calculateHeartRate(ecgValue);
46     int hrv = calculateHRV(heartRate);
47
48     Serial.print("Heart Rate: ");
49     Serial.println(heartRate);
50     Serial.print("HRV: ");
51     Serial.println(hrv);
52     Serial.println();
53
54     lastReadTime = millis();
55 }
56 }
57
58 bool isPeak() {
59     // Use the getPeak method of the PeakDetection object.
60     return peakDetector.getPeak() != 0;
61 }
62
63 int calculateHeartRate(int ecgValue) {
64     if (isPeak()) {
65         unsigned long currentTime = millis();
66         int timeBetweenPeaks = currentTime - lastPeakTime;
67         lastPeakTime = currentTime;
68
69         if (timeBetweenPeaks < MAXIMUM_TIME_BETWEEN_PEAKE) {
70             return 60000 / timeBetweenPeaks;
71         }
72     }
73
74     return 0;
75 }

```

```

77 int calculateHRV(int heartRate) {
78     rollingBuffer[rollingBufferIndex] = heartRate;
79     rollingBufferIndex = (rollingBufferIndex + 1) % ROLLING_BUFFER_SIZE;
80
81     int sum = 0;
82     for (int i = 0; i < ROLLING_BUFFER_SIZE; i++) {
83         sum += rollingBuffer[i];
84     }
85     int average = sum / ROLLING_BUFFER_SIZE;
86
87     int hrv = 0;
88     for (int i = 0; i < ROLLING_BUFFER_SIZE; i++) {
89         hrv += abs(rollingBuffer[i] - average);
90     }

```