ELL365: Monitoring Device for Measuring the Depth of Anaesthesia

Group 18
Mayank Bansal (2014EE10455)
Soham Roy (2014EE30766)

April 27, 2017

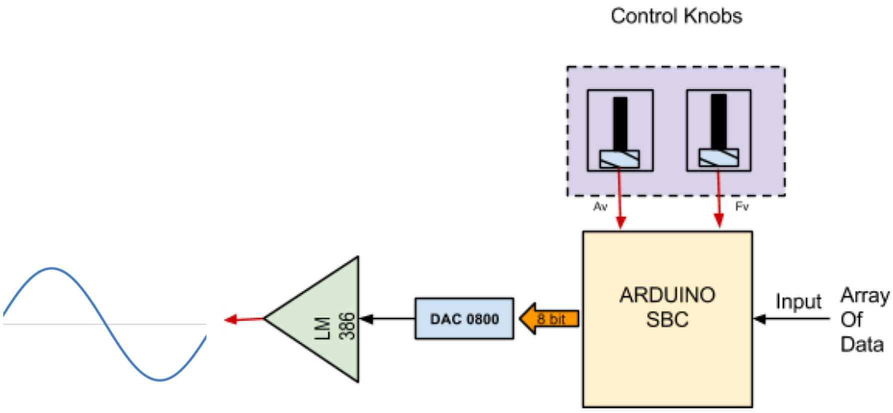
1 Aim

Our goal is to generate a sine wave on whose **amplitude** and **frequency** can be controlled and implement the whole control system on Hardware. This sine wave would be used in measuring the depth of anaesthesia.

2 Anaesthetic Depth and its Need

- **Anaesthesia** is a state of temporary induced loss of sensation or awareness in the practice of medicine, especially **surgery** and **dentistry**. It enables the painless performance of medical procedures that would otherwise cause severe or intolerable pain to a patient.
- It is used for 3 main goals:
  1. **Hypnosis**: A temporary loss of consciousness and with it a loss of memory
  2. **Analgesia**: Lack of sensation which also blunts autonomic reflexes
  3. Muscle relaxation
- **Anaesthetic depth**[3] or **Depth of General Anaesthesia (DGA)** is the degree to which the central nervous system (CNS) is depressed by a general anaesthetic agent, depending on the potency of the anaesthetic agent and the concentration in which it is administered.
- To determine **anesthetic depth**, the anesthetist relies on a series of physical signs of the patient.
- We have to **ensure adequate depth** of anaesthesia to prevent awareness without inadvertently overloading the patients with potent drugs.[1]
- Measuring DGA contributes to **tailoring drug administration** to the individual patient, thus preventing awareness or excessive anaesthetic depth and improving patients' outcomes.[2]

3 Block Diagram



4 Working

- **8 bit Digital to Analog Converter (DAC) - DAC0800**:
  - It is not possible to get analog outputs from Arduino UNO. Since we can only get digital outputs from it, this digital output needed to be converted to analog using a DAC.
  - Our DAC is **8 bit**, so for every 8-bit number it gives analog signal which is *proportional* to its decimal value. *e.g.* $11111111_2 \equiv 5$ and $00000000 \equiv 0$. Thus, it depends on the supply given to the DAC.
  - In order to generate the sine wave, we pass a **sequence** of n numbers each of 8 bits. When **converted** into analog values *sequentially*, the whole analog output signal of these n values would look like a sine wave.
- **Sine Wave lookup Table**:
  - These n numbers of 8-bit are **stored** in an integer array in such a sequence that their **plot on the Y axis** would look like a sine wave i.e. sine wave lookup Table.
  - Since the numbers are 8 bit, they lie *from 0 to 255*. To convert these integers to 8 bit Binary (8 bit Digital input of DAC, which gives output to analog) there is already an inbuilt function in Arduino i.e. *PORTD* which automatically converts an integer to binary and sets 8 digital pins (0 to 7) of the Arduino accordingly. e.g. For $252 \equiv 11111100_2$, it sets Pin 0 and Pin 1 to LOW and the rest to HIGH.
- **Setting up the Frequency**:
  - The frequency of the sine wave is decided by the time taken by Arduino to pass these n signals to the DAC. Hence, frequency control is facilitated by controlling the **sampling rate** of these n numbers to the DAC.
  - This was done by giving a **specified delay** in transmitting each 8 bit number to the DAC, such that the overall delay of sampling all **n** numbers (200 in our case) would result in a sine wave of the **required frequency**. Such that for **f Hz** frequency, Since we need to pass all 200 values in $\mathbf{\frac{1}{f}}$ *seconds* and thus each sample should pass after $\mathbf{\frac{1}{200*f}}$ *seconds* or $\mathbf{\frac{5000}{f}}$ *Microseconds*.
- **Push button**
  - On every **press**, it *increments/decrements* the frequency (i.e. set **f** accordingly) and outputs the wave of that frequency.

- Also, the value of output frequency is quantized with 0.1 within our required range of frequency i.e., 2 Hz to 6 Hz.

- **LCD**: It is used to **display** the selected frequency.

5 Cost

The cost of the Product is summed up in the Table 1.

Table 1: Cost Table

Components	Cost (INR)
Arduino NANO	300
DAC0800	75
LCD display	175
Fabrication & Packaging	250
Total Cost	**800**

Note: Arduino UNO is used to show the demonstration due to unavailability of Arduino NANO. However Arduino NANO will be used in actual product to make it comapct handy and reliable.

6 Code on Arduino IDE

```
/*
  AnalogReadSerial
 Reads an analog input on pin 0, prints the result to the serial monitor

 This example code is in the public domain.
 */
#include <LiquidCrystal.h>
LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);

int sine[201]={128,132,136,139,143,147,151,155,
159,163,167,171,174,178,182,185,
189,192,196,199,202,206,209,212,
215,218,220,223,226,228,231,233,
235,237,239,241,243,245,246,247,
249,250,251,252,253,253,254,254,
255,255,255,255,255,254,254,253,
253,252,251,250,249,247,246,245,
243,241,239,237,235,233,231,228,
226,223,220,218,215,212,209,206,
202,199,196,192,189,185,182,178,
174,171,167,163,159,155,151,147,
143,139,136,132,128,123,119,116,
112,108,104,100,96,92,88,84,
81,77,73,70,66,63,59,56,
53,49,46,43,40,37,35,32,
29,27,24,22,20,18,16,14,
12,10,9,8,6,5,4,3,
2,2,1,1,0,0,0,0,
0,1,1,2,2,3,4,5,
6,8,9,10,12,14,16,18,
20,22,24,27,29,32,35,37,
40,43,46,49,53,56,59,63,
66,70,73,77,81,84,88,92,
```

```
96,100,104,108,112,116,119,123};
float f=2.0;
int st=0;
void setup()
{
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("FREQUENCY OF SINE WAVE");

  pinMode(0,OUTPUT);
  pinMode(1,OUTPUT);
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(9,INPUT);


}

void loop()
{
  st = digitalRead(9);
  if(st==HIGH){
  f+=0.1;
  lcd.setCursor(0,1);
  lcd.print(f);
  }

  for(int i=0;i<200;i++)
  {

 PORTD=sine[i];

  delayMicroseconds(int(5000/f));
  }
}
```

7 Observations

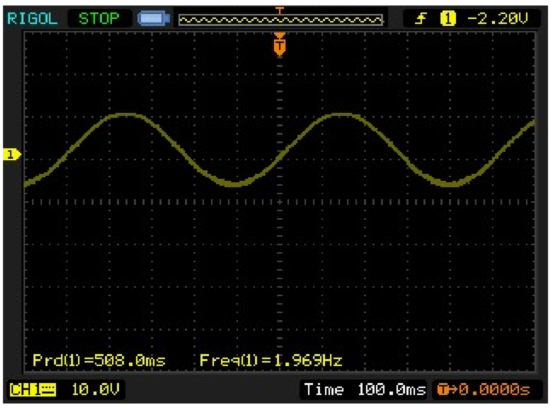


Figure 1: Sine wave of Frequency 2 Hz

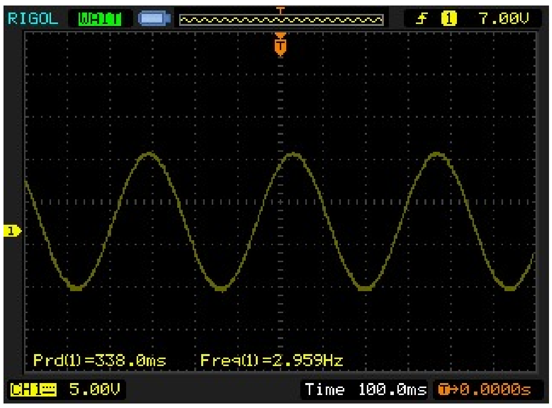


Figure 2: Sine wave of Frequency 3 Hz

- We took *n=200*, which is fine for our frequency range (2-6 Hz).

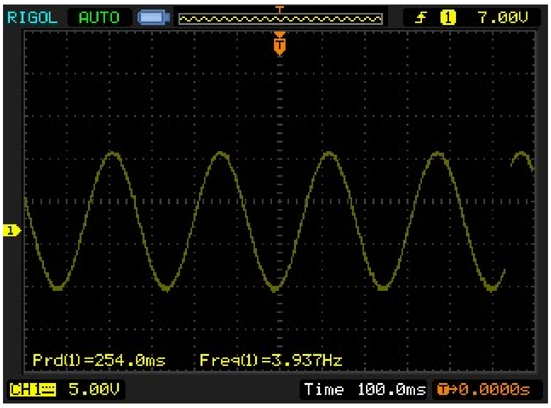


Figure 3: Sine wave of Frequency 4 Hz

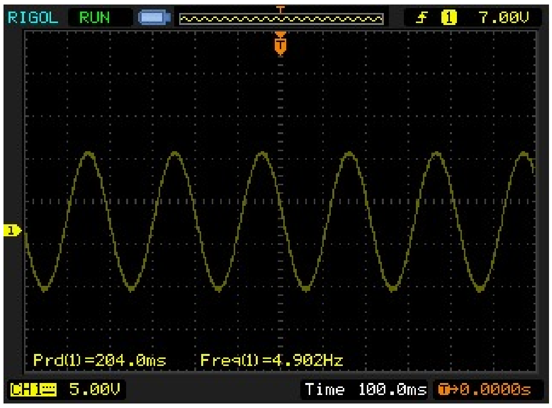


Figure 4: Sine wave of Frequency 5 Hz

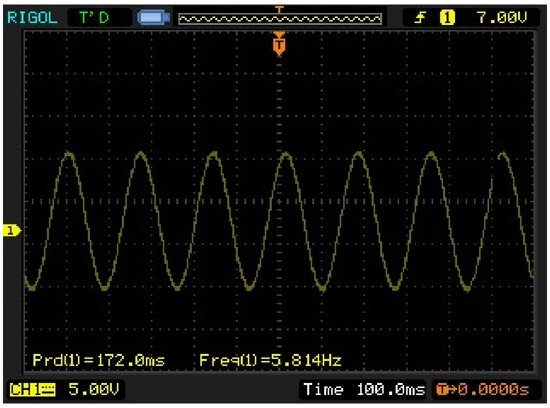


Figure 5: Sine wave of Frequency 6 Hz

- For higher frequencies, we need more samples, i.e. a higher value of n.
- Due to **finite resolution**, the output is a **discrete** sine wave and not a continuous one.
- For a high number of samples in a small time, it looks like a continuous sine wave on the screen.
- Upon **zooming** it looks like a **stairstep plot** of a sine wave.

8 Applications

- The device produces a sinusoidal voltage of adjustable amplitude and frequency. By using a **potentiometer** or LM386 Operational Amplifier this voltage can be converted to a sinusoidal current of a required amplitude and the same frequency.
- This **sinusoidal current** of low frequency and low amplitude, generated using metal electrodes can be passed through the human body. This would allow monitoring the **depth measurement** of ***anaesthesia***.
- The final fabricated product will be used by a doctor at **AIIMS** who is collaborating with Prof. Rahman.

References

[1] H. Kaul and N. Bharti. Monitoring depth of anaesthesia. *Indian Journal of Anaesthesia*, 2002.

[2] B. Musizza and S. Ribaric. Monitoring the depth of anaesthesia. *Sensors*, 2010.

[3] D. D. Rani and S. Harsoor. Depth of general anaesthesia monitors. *Indian Journal of Anaesthesia*, 2012.