Assignment No. B-1

Roll No:

Aim :

Write an application to and demonstrate the change in BeagleBoard/ ARM Cortex A5 /Microprocessor /CPU frequency or square wave of programmable frequency.

Software Required :

- Linux Operating System
- GCC Compiler.

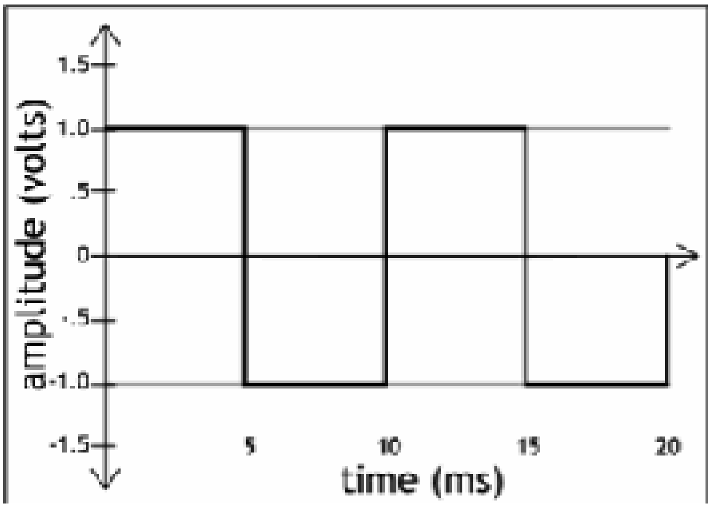
Hardware Required :

- Beaglebone Black/ ARM Cortex Processor
- Interfacing cables and CRO

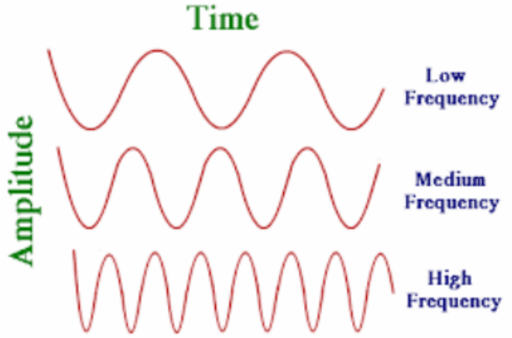
Theory :

A square wave is a non-sinusoidal periodic (which can be represented as an infinite summation of sinusoidal waves), in which the amplitude alternates at a steady frequency between fixed minimum and maximum values, with the same duration at minimum and maximum. The transition between minimum to maximum is instantaneous for an ideal square wave .

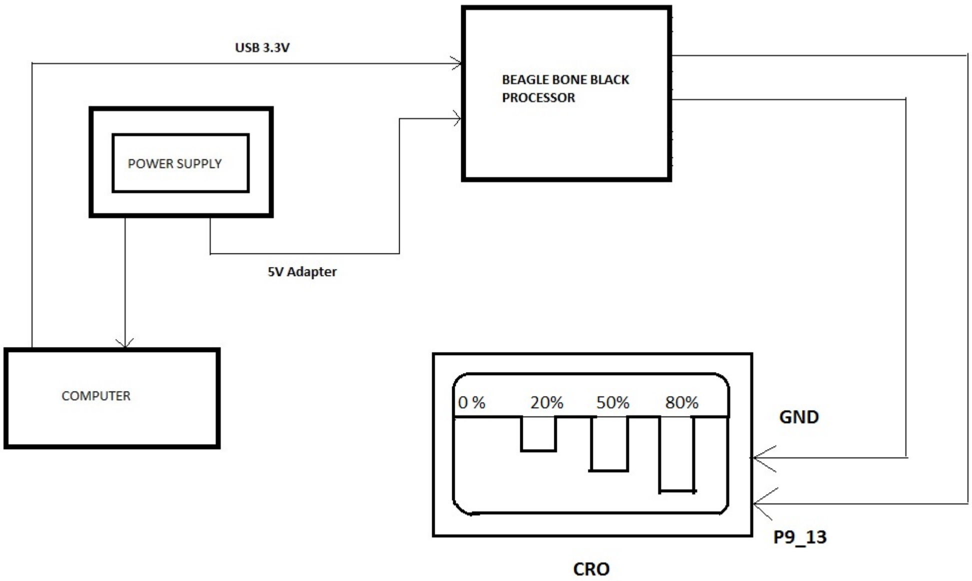
A duty cycle is the percentage of one period in which a signal is active . A period is the time it takes for a signal to complete an on-and-off cycle. It is the proportion of time during which a component, device, or system is operated. The duty cycle can be expressed as a ratio or as a percentage. Suppose a disk drive operates for 1 second, then is shut off for 99 seconds, then is run for 1 second again, and so on. The drive runs for one out of 100 seconds, or 1/100 of the time, and its duty cycle is therefore 1/100, or 1 percent.



Frequency describes the number of waves that pass a fixed place in a given amount of time Example: if the time it takes for a wave to pass is is 1/2 second, the frequency is 2 per second. If it takes 1/100 of an hour, the frequency is 100 per hour. The unit of frequency is Hertz.



Interfacing Diagram



Algorithm

1. Set up the connection as shown in block diagram.
2. Give +5V AC supply to BeagleBoard Black.
3. Write and Design an application program in gedit to generate a Square wave.
4. Compile application program .
5. Provide power supply to CRO and interface CRO with BeagleBoneBlack.
6. Finally we can see Square waves is generated on CRO.

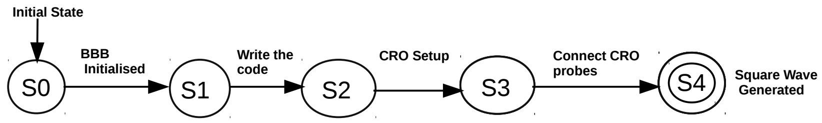
Mathematical Model

Let S be a set such that

S={s, e, i, o, f, DD, NDD, success, failure}

s= initial state

e = end state



i= input of the system.

o= output of the system.

f= functions

DD-Deterministic Data it helps identifying the load store functions or assignment functions.

NDD- It is Non deterministic data of the system S to be solved.

Success-Square Wave generated on CRO

Failure-Desired outcome not generated or forced exit due to system error.

States: { S0 ,S1 , S2 , S3 , S4 , S5 }

S0: initial State (Power supply)

S1: BeagleBone Black initialisation

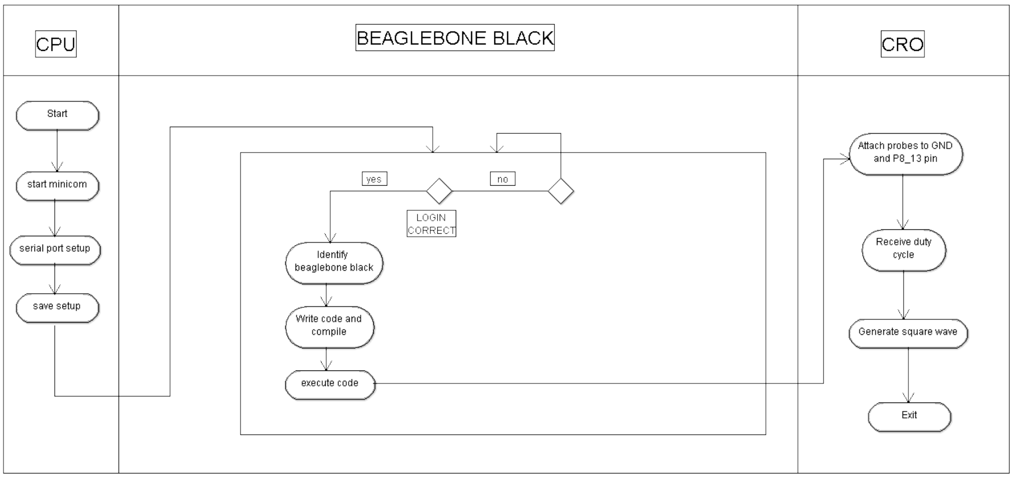
S2: Monitor Display editor (Write and Design application using gedit)

S3: CRO initialised

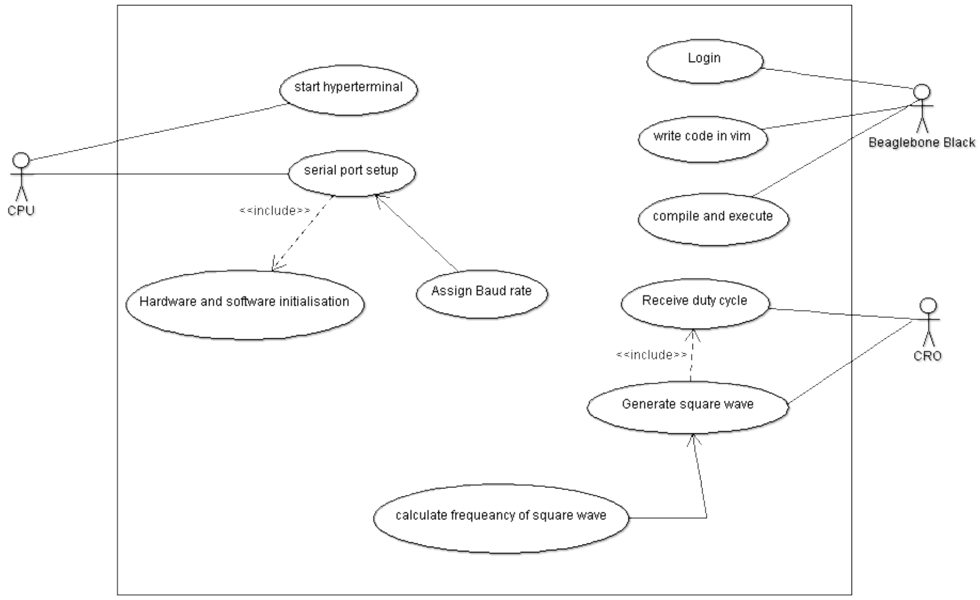
S4: Square Wave Generated. (Final State)

UML Diagrams :

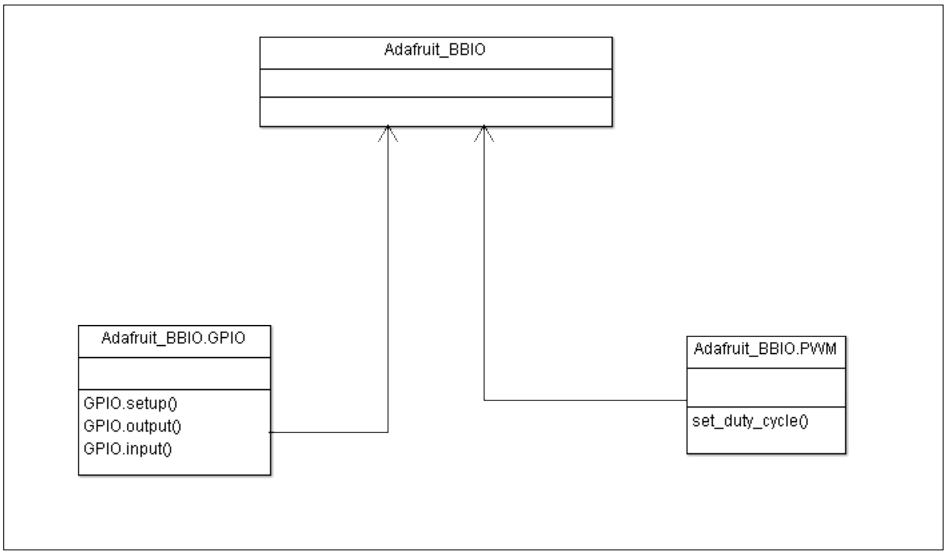
Activity Diagram



Use-case Diagram



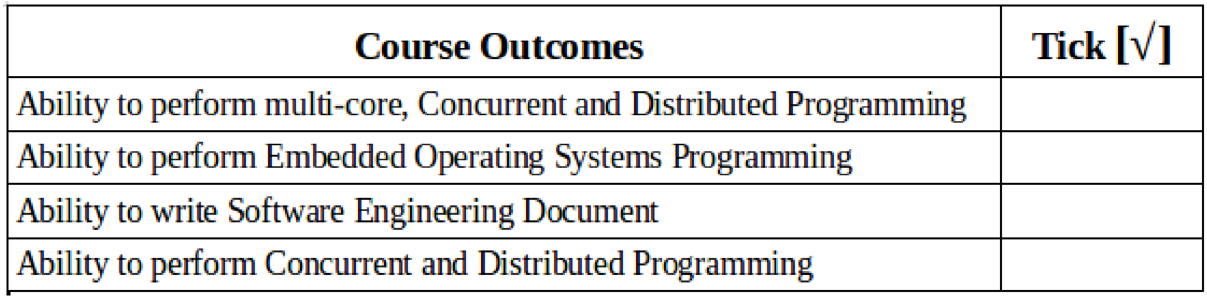
Class Diagram



CONCLUSION :

Hence, we have successfully developed and demonstrated Robotic Arm application using stepper motor with Beagleblack bone board..

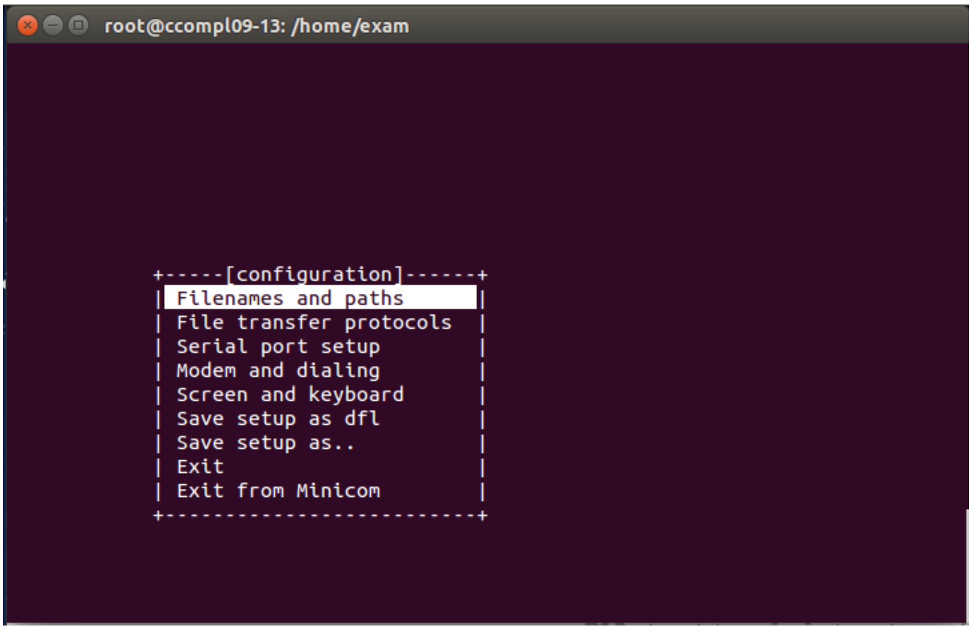
Course Outcomes :



Course Outcomes	Tick [√]
Ability to perform multi-core, Concurrent and Distributed Programming	
Ability to perform Embedded Operating Systems Programming	
Ability to write Software Engineering Document	
Ability to perform Concurrent and Distributed Programming	

Output (Screenshots)

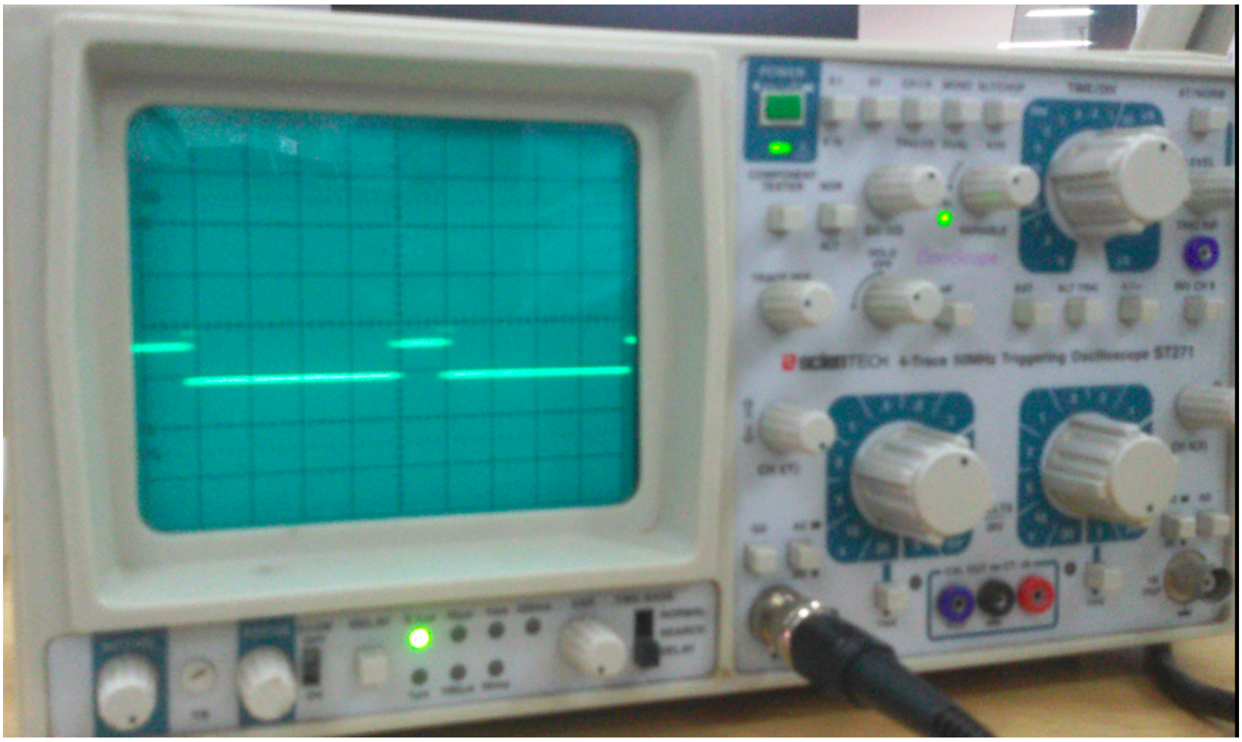
Minicom Terminal



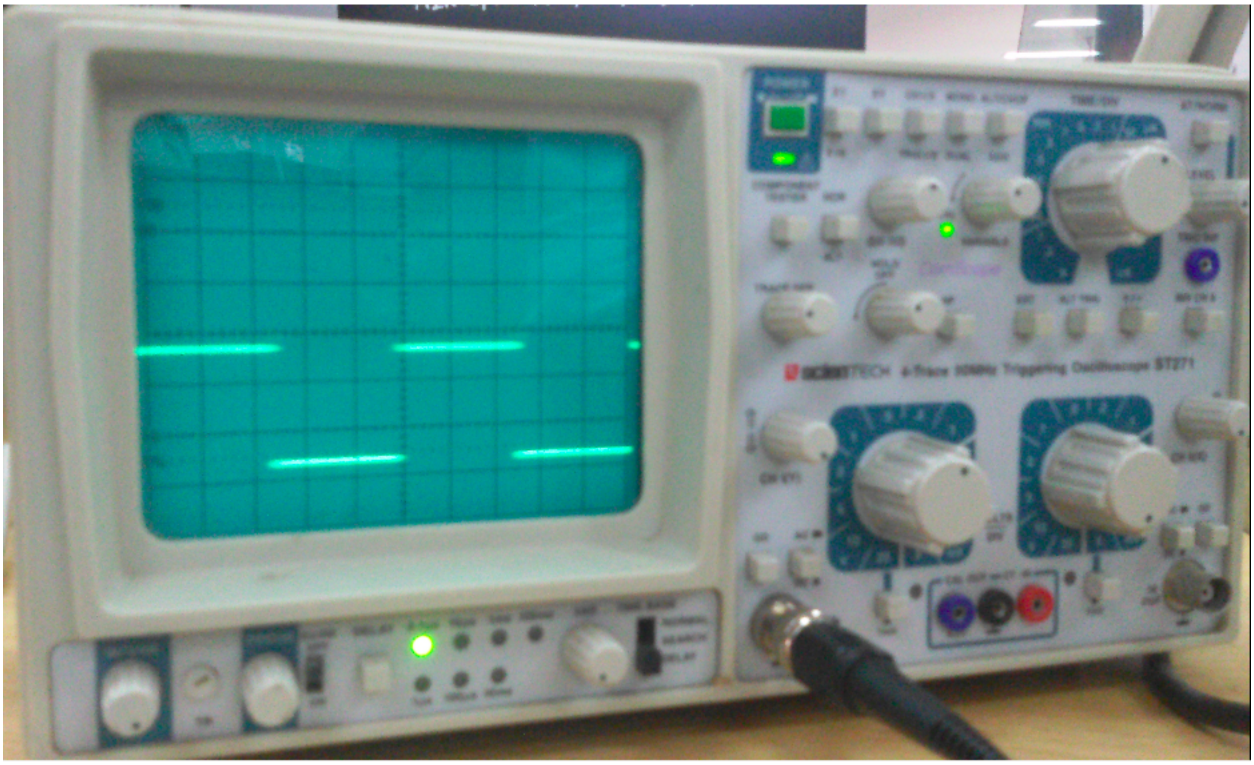
root@ccompl09-13: /home/exam

```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

CRO Output 1:



CRO Output 2:



Code :

```
import Adafruit_BBIO.GPIO as gpio
import Adafruit_PWM as pwm
import time

pwm.start("P8_13",50)
while 1:
pwm.set_duty_cycle("P8_13",0)
time.sleep(0.5)
pwm.set_duty_cycle("P8_13",20)
time.sleep(0.5)
pwm.set_duty_cycle("P8_13",50)
time.sleep(0.5)
pwm.set_duty_cycle("P8_13",80)
time.sleep(0.5)
pwm.stop("P8_13")
pwm.cleanup()
```