Arduino Programming Challenge (APC) Finals.

Elecfest, Events, Shaastra-2020.

TOTAL POINTS: 65 + 15(bonus)

Attempt the bonus question only after you complete the other 4 questions.

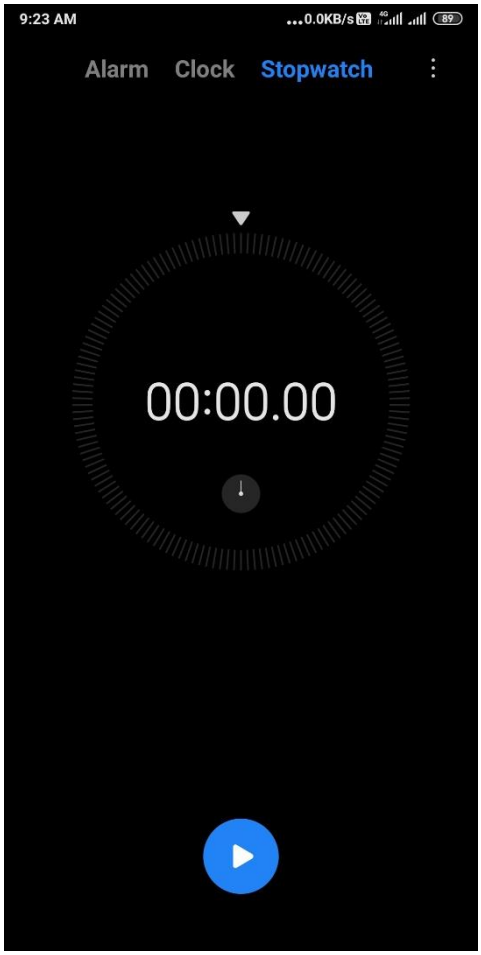
1 Building a stopwatch using the internal clock of Arduino

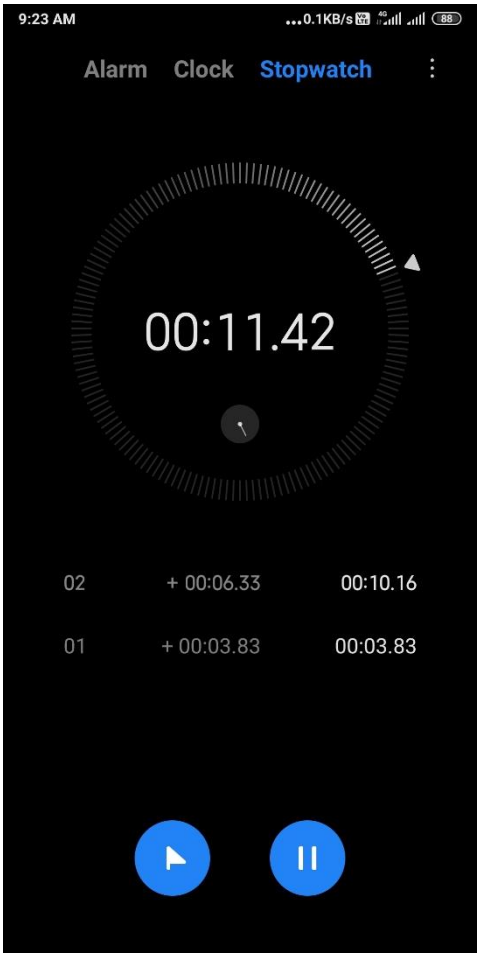
Point:20

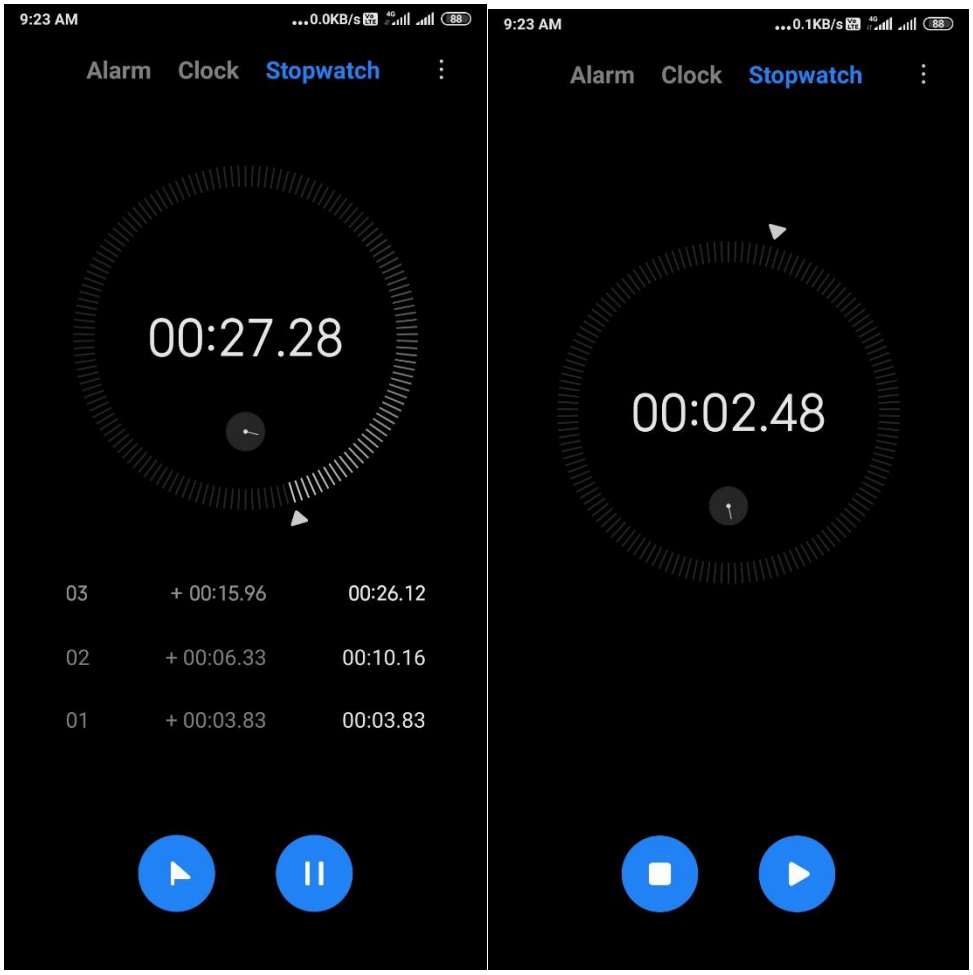
INTRO : A **stopwatch** is a handheld timepiece designed to measure the amount of time that elapses between its activation and deactivation. A large digital version of a stopwatch designed for viewing at a distance, as in a sports stadium, is called a **stopclock**. In manual timing, the clock is started and stopped by a person pressing a button. In fully automatic time, both starting and stopping are triggered automatically, by sensors.

PROBLEM : You are asked to build a stopwatch which has 4 switches. One for each Start, Pause, Flag, Show the Previous Flags and the Completed Time. This should work in the same way any stopwatch in a smart Phone would work. It should start once the Start button is pressed noting the time in the format min:sec:cents. The Pause button should pause the timer till the Start switches again. Whenever Flag is pressed, the time at that instant is marked(as in it is stored in the memory). The SPFCT(Show the Previous Flags and the Completed Time) when pressed should show the flags that have been marked till now and the time ran till now. All of this should be printed on the Serial Monitor.
Maximum no. of flags should be 4.

FOR EXAMPLE: Your Serial Monitor should show similar output along with the time difference between consecutive flags.


9:23 AM
...0.0KB/s
Alarm
Clock
Stopwatch
00:00.00


9:23 AM
...0.1KB/s
Alarm
Clock
Stopwatch
00:11.42
02
+ 00:06.33
00:10.16
01
+ 00:03.83
00:03.83

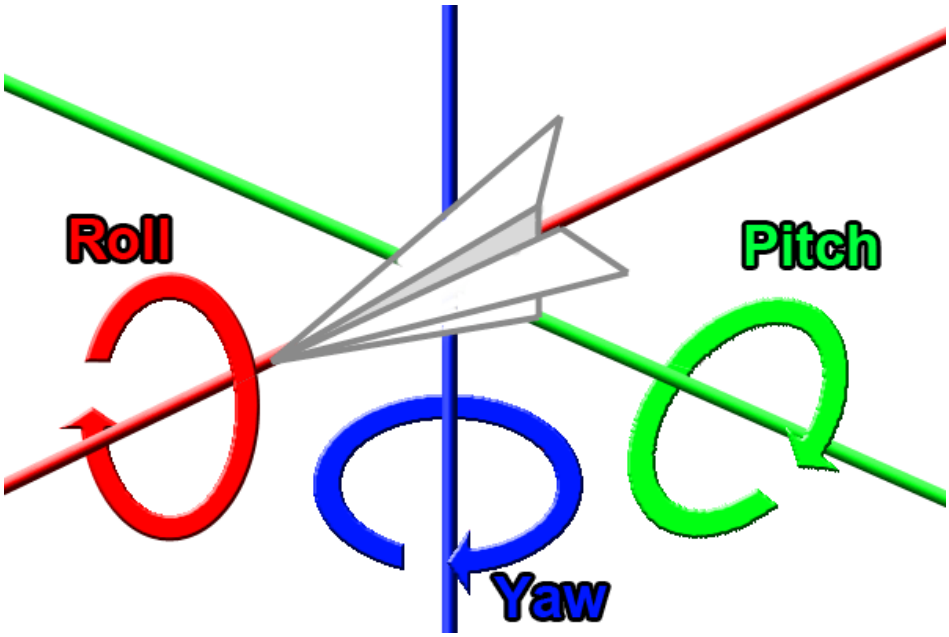


Remark : Update the flags when a flag is recorded after the 4 flags have been marked and you are left with no space for the new flag.

2. GESTURE CONTROLLED BOT SIMULATION USING IMU (GCBS): **Points: 20**

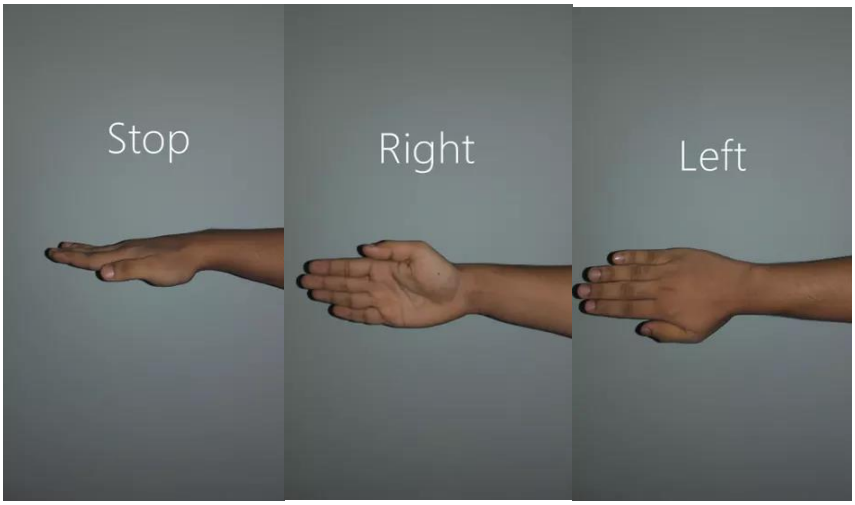
INTRO : The MPU6050 has an embedded 3-axis MEMS gyroscope, a 3-axis MEMS accelerometer. It is very useful for lot of motion detecting. Almost all smart phones now have accelerometers. You have definitely played motion games like Temple Run in your mobile where

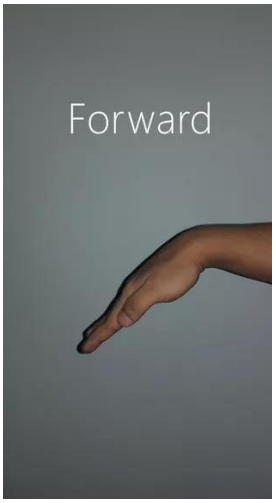
the character in the game moves left and right when you tilt your phone left and right respectively. This is done with the help of accelerometer.



PROBLEM :

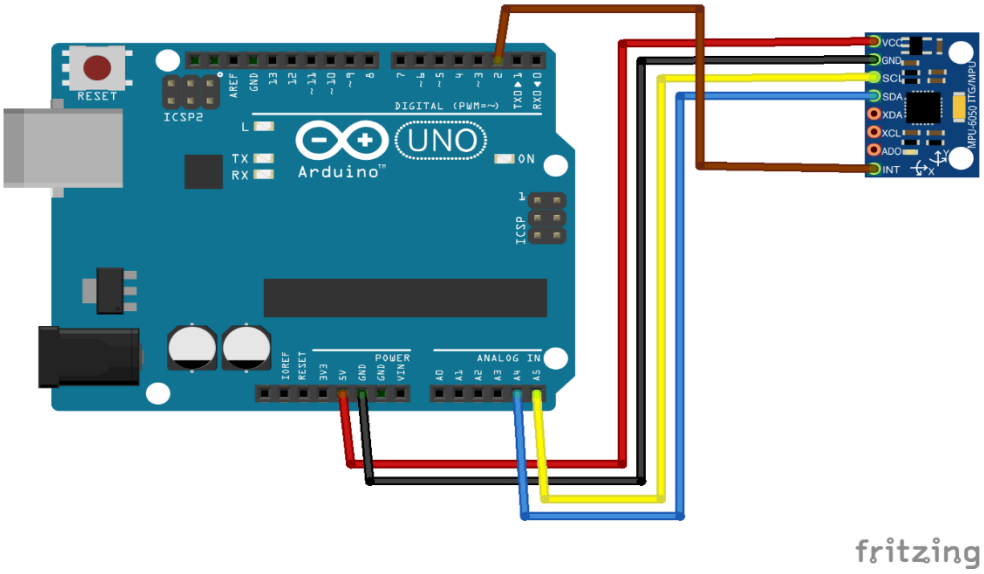
Being tired of controlling robots with a joystick, you wanted to control a bot with your hand. So, you planned to design a bot which recognizes five gestures: Forward, Backward (opposite to forward), Left, Right and Stop. You will get better ideas if you check the images in the next page:





To achieve this you can use the GY521 - MPU6050 IMU and attach it with the palm using gloves. After obtaining the raw accelerometer and the gyroscope values from the IMU (inputs), **devise a post-processing algorithm**, a mathematical model (like a Machine Learning model or simple if-else statements), which takes in the sensor values and outputs whether the gesture is forward or backward or left or right or stop in **serial monitor**.

Remarks: Use Jarzebski library for extracting data from the IMU



3 Finding all possible remainders when N is divided by all positive integers from 1 to N+1 Point:10

PROBLEM : Given a set of 5 numbers in an array (in the format as mentioned in "FOR EXAMPLE" below) as input from serial monitor, output a arrays consisting of the required remainders in ascending order separately for all the numbers and print the Array in the Serial Monitor. Also, use **millis()** to measure the overall time from beginning time of the code till the end. The above time length will be used as a tie-breaker. The team with the least time length in case of a tie will be announced as the winner.

FOR EXAMPLE: If the input is [3,11],
then the output is [0,1,3]
[0,1,2,3,5,11].

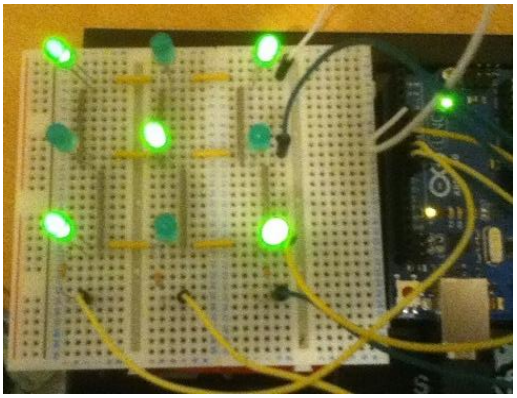
Remark : We know that you can solve this problem :) But we want to test you based on algorithm and the time taken to solve. The maximum number of digits in an element of the sample input array will be 4.

4. **Alphabets display using LED's and PWM (LED): Points : 15**

INTRO : A Light Emitting Diode (LED) is a semiconductor light source that emits light when current flows through it. The color of the light is determined by the energy band gap of the semiconductor. Modern day LED diode TV screens uses RGB leds combination (and controlled by a powerful microprocessor) to display moving frames on the screen.

PROBLEM : Given 9 LEDs of different colors and two potentiometers, place the LEDs in a 3*3 matrix. These 9 LEDs are to be used to display the "English Alphabet" .Which is given as the input using one of the potentiometers i.e., this potentiometers reading is divided into 26 parts and each consecutive part corresponds to one consecutive alphabet. The other potentiometer is used to dim and brighten the LEDs. Take voltage and current requirements of LED's also into consideration.

FOR EXAMPLE : If the potentiometer for selecting the Alphabet is at it's 24^{th} division, the necessary LEDs in the 3x3 matrix should glow such that it displays the letter 'X'.



HINT : Syntax for PWM in Arduino IDE: `analogWrite(pin, value)` where:

1. pin: the pin to write to [allowed data type: int];
2. Value: the duty cycle between 0 (always off) and 255 (always on) [allowed data type: int].

Bonus:

5 **Making an Alarm Clock** **Point:15**

INTRO : An **Alarm Clock** (or sometimes just an **alarm**) is a clock that is designed to alert an individual or group of individuals at a specified time. The primary function of these clocks is to awaken people from their night's sleep or short naps; they are sometimes used for other reminders as well.

PROBLEM : You are supposed to make an Alarm Clock using an external RealTimeClock(DS3231), 4 switches, 1 buzzer and the Built-in-LED.4 switches One for each Set-Menu, +, -, Switch Alarm(off). Use the '+' and '-' buttons to increment the hrs, day, year , month, min while setting alarm and also for selecting which of the time segment to set.
For the alarm sound , use the frequencies 880 and 698 with a delay of 0.3 secs between them**. You are supposed to use the**

RTClib.h library. While saving in the alarm time the message "SAVING …" "IN PROGRESS…" and "DONE".

The alarm will be set for 2 minutes from now and then checked for the buzzer.