Name = Sameep Vani

Roll Number = AU1940049

Major = B Tech CSE

Course Code = ECE 302

Course Name = Embedded Systems Design

Section = 1 , (PART A)

Undertaking -

I certify that I have not violated the University code of conduct during this examination.

S. N. Vani

03/10/2021

□ + □ + □ + □ + □ = □

A1→ Main ~~the~~ Criteria = Computational needs. ~~How~~ This includes efficiency, cost etc.

Sub criteria (a) = Speed. This should be high for a good micro controller.

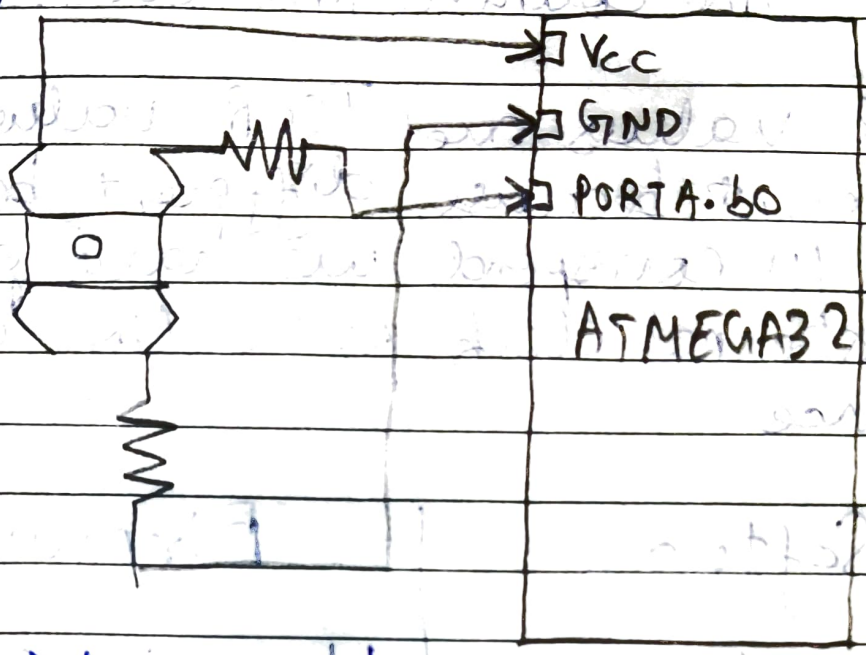
Sub-criteria (b) = Power Consumption. This should be not too low and not too high. Adequate power consumption should be there.

Sub-criteria (c) = Packaging. The ~~space~~ required should be low. ~~Easy~~ It should be easy to assemble etc.

Sub-criteria (d) = Amount of RAM & ROM on chip.

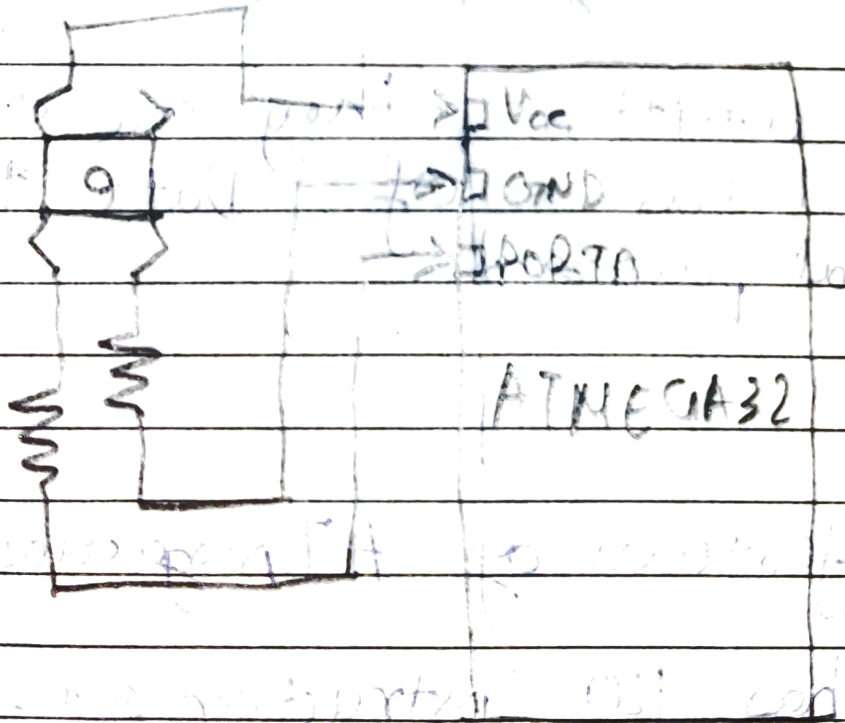
A2

2(a) Here I am connecting PORTA.0 to Atmega32 and drawing only necessary PINS.



As it can be seen when button is not pressed, 0 is passed in PA0. and vice versa

(b)



Here, in idle state, V_{cc} is connected to PORT and hence '1' is passed.

A3 (a)

Timer 0	Timer 2
① Is not used as real time counter	Is used as real time counter
② TCCR values are different because 110 & 111 correspond to external clock source	TCCR values are different because we have 2 more prescaling options

(b)

Software	Firmware
① High-level program which does not affect hardware even if changed.	Low level program without which hardware functionalities won't work
② It is written in C and are "actual programs"	They are a type of Data Structure

(c) Two features of ATmega are:-

① It has 120 instruction set & lots of peripheral capabilities.
② Its program memory ranges from 4K to 256K bytes.

(d) Two types of ES →

(1) Stand-alone →

- Designed to perform specific task repetitively.
- Non-reactive to env.

(2) Real-time →

- Monitors environment where it is installed
- Required to respond in time. to a request.

A4

```
#include <avr/io.h>

void main()
{
    DDRA = 0xFF; // Output Mode
    DDRB = 0xFF;
    int x = 49; // Packed BCD in decimal
    int temp1, temp2; // Some var.
    temp1 = x & 0x0F; // Get last 4 bits
    temp2 = x & 0xF0; // Get first 4 bits
    temp2 = temp2 >> 4; // Shift to LSB
    temp1 = temp1 | 0x30; // add 3 to MSB
    temp2 = temp2 | 0x30;
    PORTA = temp1; // Display
    PORTB = temp2; // Display
}
```