DRISHTI

A Revolutionary Concept

Where “We” comes before “Me”!

Makerspace: 5.0

Project Name: Digital Clock

Date: 30.03.2021

Installation : We were provided the software links of Proteus Professional and Atmera Studios. The softwares were downloaded from there. Few Team mates faced some issues in Atmera Studios as link of offline installer was not clickable. But direct link was provided and issue was solved.

Next we disscussed about creating a new Project file in Proteus. Using AVR and AtMega32, in schematics we were supposed to have Atmega 32 Port but yet there are some issues with that.

Issues Resolved with Run As Administrator Feature while opening the software.

We started the discussion with what pins are in an ATMega32. We concluded that there are 40 pins. 32 of them are grouped in 4 slots of 8 each. The remaining are:

RESET: Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. . Shorter pulses are not guaranteed to generate a reset.

XTAL1: Input to the inverting Oscillator amplifier and input to the internal clock operating circuit. XTAL2 Output from the inverting Oscillator amplifier.

AVCC: Is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.
AREF: Is the analog reference pin for the A/D Converter.

For the this particular project we do not use
XTAL1, XTAL2, and GMD.

Next, we discussed,

1. What are the pins GND and VCC for?
2. What is difference between port A, port B so on.
3. What really is the function of pins and how is data transmitted?

Answers each in order

1) GND and VCC are pin number 10,11 and 30,31. Usually the pins not being used are shorted by soldering them. This is done on a Circuit Board since high temperatures damage the IC. It does not matter if say Ground (11) is connected to VCC (30). The coming discussion covered a vast area like.
   A) Suppose we have a 12V batter, how will be achieve 5 Volts?
   Answer: By using a Potential Divider Circuit.

Correct Answer: Voltage Regulating IC (Model LM 7085)
B) Will there be any impact on Pulse circulating in the microcontroller if we connect say, 11 and 30?
Answer: No there will be no difference. The ground pin simply provides a safety mechanism.

2) Port A is an Analog Input Port. Port B is a digital input port. Port C is a non-reprogrammable port. Hence it is usually not used. Say you are using port A and B, then the remaining ports not in use are shorted.
3) Edited: 7.04.2021: By what we can make out till now pins serves as connections between IC's and Microcontroller. While writing code, **DDRx** must be used to define which pin is high and while is low.

DDRx: It stands for Data Direction Register. It directs/commands which pin a a specifed port **'X'** will give you output and which one will give you input.

Then we started Discussing I^2C communication protocol. Terms involved were Slave, Master, UART, USART (Universal Synchronous Asynchronous Receiver and Transmitter) , etc...

Conclusion: Master (Microcontroller) and Slave (RTC).

Reference: (1)

Date : 31st March 2021

Today We Learned about several types of Communication Protocols.UART, USART, I2C, SPI etc. and their types such as Inter or Intra. We will be using I^2C for our project. Reason being, it is easy to connect multiple devices just by using 2 wires of I2C (SDA and SCL). In case of other we may need many ires and it will make the circuitary very very complex.

Further Master and Slave Devices were addressed and their internal data sending mechanism was also explained.

Finally we learned about new device called as Voltage Regulators

which eventually help in reducing the Input Voltage (12V to 5V). There are many types of Voltage Regulators available, we learned

about LM7805. In this there are 3 pins. One for Input , Middle one for Ground, Last one for Output.

We also add two Capacitors (In parallel) in Voltage Regualtors Circuit to reduce noise. Or it may interefere and cause

disruptions. The two capacitors are generally in 1:10 ratio.

Some more info in Voltage Regulating IC's: Link
1. The last two digits of the model say, 78**05** **or** in general 78**xx** represent the voltage output of the Regulator IC.

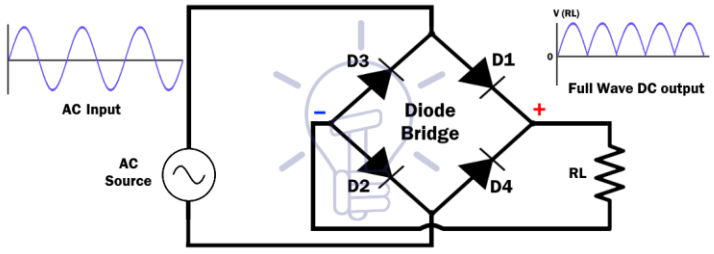
2. Voltage IC's have 3 pins. Output/Voltage, Input, Ground/Low.

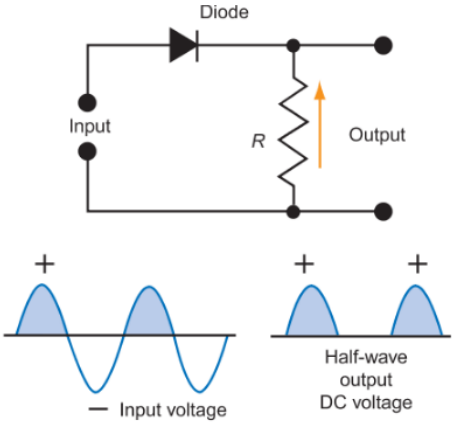
Dropout Voltage: The dropout voltage for any regulator states the minimum allowable difference between output and input voltages if the output is to be maintained at the correct level. For example, if a LM7805 regulator is to provide 5V at its output, the input voltage must be no lower than 5v +2.5V =7.5V.
Reference : (1)

A new Question was Posed.

Q) How will we convert Say 12V AC to 5V DC.

Answer: We will use a rectifier to convert AC to DC first.





Then the voltage can be regulated using a Voltage Regulating IC.

Date : 1st April 2021

We Learned about I^2C Protocol. The complete block diagram was explained. From Start bit and 7-bit address from Master Device then moving through the Read/Write bits followed by Ack/NAck bits and finally about Stop bit in detail.

Embedded C libraries were discussed. And some code related doubts were addressed.

Introduction to '>>' and '<<' BITWISE Operator in Embedded C.

Date : 3nd April 2021

Diodes: - They function to restrict current in 1 direction. In a circuit they provide safety by shorting

Example: A Li Po Battery is accidentally connected to opposite terminals. Diodes stop the flow of current in opposite direction thereby preventing it from exploding

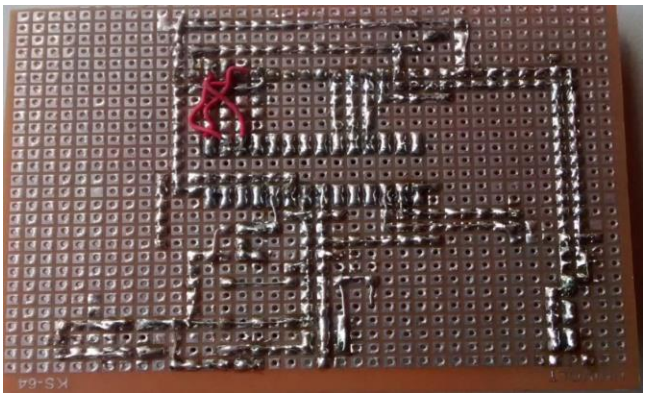
In code , PORTC = 0xFF Activates all the pins of PORT C so we can get output by connection LED to any Pin.

Note: Port C is not usually used since it is not reprogrammable.

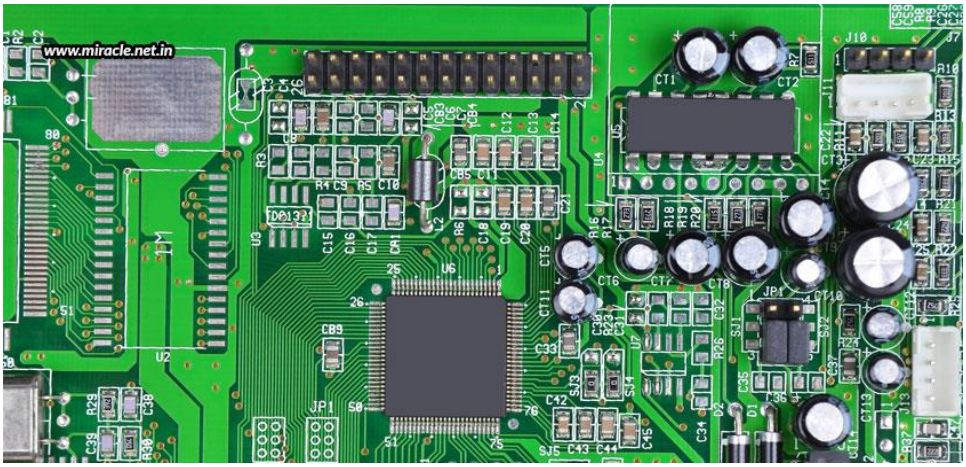
Date : 6th April 2021

Some Useful Short forms:

GCB: General Circuit Board. It's a less complicated IC having limited Components



PCB: Printed Circuit Board. It's a much more complex IC having wide range of components, internal buses etc.



We simulated LED Blinking Codes for all team members. One Problem which we faced was even after correct code and Circuit Connections the LEDs weren't blinking. Most Probably the reason must be in the setup of Project in Proteus the PCB Schematic was selected that might have caused issues.

<u>Also, there are two types of errors while writing code :</u>

Compile Time Error: Error which occurs when the program is built. It may be a syntax error or a forgotten semicolon or library.

Run-Time Error: Such Types of errors are most difficult to find since they indicate fault in logic.

<u>Example</u>: For LED Blinking. You've set A1 as the output pin according to your code, but connected A0 in the simulation. The LED will not blink but the code might not show any built errors.

If sometimes code is not running try the following things: -

A) Make a new project both in ATMEL and Proteus, making sure you are selecting the right options.

B) Include the libraries if not included

Ifdef, F_CPU etc.

C) Use code/setup from someone, who you know has a running project/trial.

Learned new thing, instead of using Ground, we connected the Pin to other PORT and made it low output type to allow the potential difference among two pins and thus LED was made to blink.

We, Saw GCB, PCB, Soldering Guns, Jump Wires. The major drawback of using Jump Wires is that in case of transportation the jump wires may tend to break or loose in its connection. So, using them in minimum number is better option. (It is a common challenge to use minimum number of Jump Wires). Soldering Metal drops if connected by mistake then we are supposed to de-solder the whole GCB to re-solder it. Using PCB is beneficial because we can add so many elements on it as compared to GCB.

It is generally, better practice to use Binary Codes rather than using Hexadecimal Codes. Reason being if we need particular pins to be high or low, we can describe it with 0s and 1s but We do not know Hexadecimal codes for the same.

Date : 7th to 9th April 2021

LED Code was left for some members so we made attempts to solve the issues and finally completed code. Then we discussed about presentation that was supposed to be shown in Update Meet. Part by Part RTC being specific DS1307 was also discussed. The whole procedure of Data storing to Data Passing by RTC was discussed. The common doubt raised was about the Crystal Capacitor, The Oscillator circuit that matches the frequency in case of Discrepancies due to environmental facture like Temperature or Wind Pressure or Technical factors like Varying Resistor Loads. So, We learned it was necessary in practice to add CYSTAL Capacitor to the Circuit along with RTC and LCD.

Date : 10th April 2021

Update Meet

Date : 12th April 2021

We were assigned the task of write the whole code of I2C without using any kind of libraries. For that, It was necessary to know about the registers and Data flow provided into the register logics. We utilized the day for learning the Register types and functions for each of them.

While writing code we were mistaken to consider Registers as independent variables but they turned out to be the predefined values of AtMega avr library.

Date : 13th April 2021

The code of I2C was made by us but we missed the Master Transmission functions to get called in main function. We, solved few confusions that were prevailed over the code and Controller. Some of them were:

1. BCD to Binary and Binary to BCD Conversion.
2. Some misconceptions of I2C Libraries.
3. How RTC would be deploying data and pass it to LCD

From then, we learned the conversion of Hexadecimal to Binary.

Date : 14th April 2021

Today we made another attempt to complete the code regarding I2C Control functions. There was a major doubt with regard to Error part being executed inside the if statement. The statement was supposed to be converted into the while statement so it was bit confusing about the happening of the execution. Further we were explained about the Function callings and code of I2C finally made by mentors. We learnt to over write the values in control register as and when required. The left shift operator were implemented so as to set the pins in required output forms. Datasheet was prime reference and all the understanding got cleared more than we had assumptions earlier for the same.

Some Unnecessary functions were supposed to be elimated from our code as they were already predefined. And some minor modifications were required.

Next, We discussed how we can change from Master Transmission to Master Receiver Mode. The emphasis must be put on the point that, It is impossible to go to the Master Receiver Mode without Bypassing Master Transmitter mode. Also, We are supposed to add MT and MR both modes in start code itself and initiate the control register values both time we want to start with any of these mode.