



Digital Clock

Jarnu Girdhar (160020028, jarnu.girdhar.16002@iitgoa.ac.in)

Avyta Bansal (160010024, avyta.bansal.16001@iitgoa.ac.in)

Ishan Singhal (160010016, ishan.singhal.16001@iitgoa.ac.in)

Rohan Teja (160010021, rohan.teja.16001@iitgoa.ac.in)

Introduction

In this project we implemented a 24 hour format digital clock using 4 digit Binary-Coded Decimal (BCD) Counter. The clock will not only display the current time but will have additional functionalities as well such as Stopwatch, Colour Change, Show Temperature, etc.

System Overview

A block diagram of the system is given in Fig. 1.

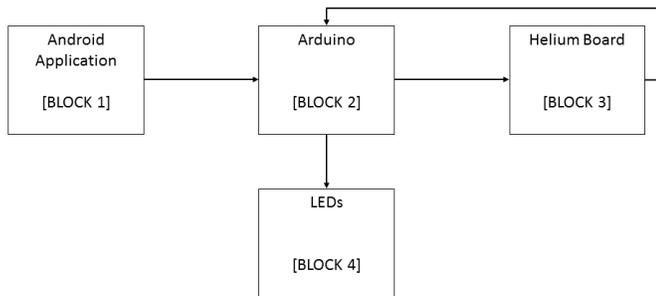


Fig. 1: Block Diagram.

The elements of the digital clock are Android Application, Arduino (I/O), Helium Board (I/O) and LED's.

1. Android Application (BLOCK 1)

- Contains buttons to control the clock manually.
- Buttons transfers the input signal via bluetooth module to arduino.

2. Arduino (BLOCK 2)

- Fetches the input from the android application.
- Converts input in binary format and outputs it to the helium board.
- Fetches input from the helium board after the counting process.
- According to input received, activates the led functions and lights the LED's.

3. Helium Board (BLOCK 3)

- Takes input from the arduino and starts the counting process according to the code.
- Gives output and passes it on to the arduino for processing.

4. LED's (BLOCK 4)

- LEDs will show the desired time output.

Implementation Details

We made an Android Application to control the clock functionalities. As we press the button Show Time on our Application, it fetches the current time of the phone, sends it to the Serial Monitor of Arduino which is then converted to a 4-bit binary number for each digit. For instance, if the current time is 12:13PM, we have written code such that it detects whether it is PM or AM and then convert it accordingly to 24-hour format. Now, the Arduino code converts each digit to 4 bit binary number, i.e. 0 -> "0000", 1 -> "0001", 2 -> "0010", 3 -> "0011" and this is given as input to helium board one by one, the enabling of load and reset button is internally controlled by code using delays according to the clock timing. These bits are stored in D-latches and are then fed to the counter. Now, to show time physically, we have implemented a 4 bit 7 segment display using WS238B LEDs and Arduino, the LEDs are controlled using Arduino code, the output binary number from counter is given as input to 4 Arduino pins one by one which are read using `digitalRead(pin)` (command of Arduino) and then accordingly the LEDs are lit up. The process is quick because of high frequency clock available in Arduino board (8MHz) which shows all the digits at once without visible delays.

Results

The buttons on the android application are functional and are as follows:

Connect: Connects the android phone with Arduino module to control the clock.

Show Time: Displays the time on clock. Fetches time from Real Time Clock module.

Stopwatch: Starts time count up on the clock.

Show Temperature: Shows current temperature from Android Phone.

Change Brightness: Adjusts brightness of LEDs of clock.

Change Color: Adjusts color of LEDs.

So the digital clock is working as we expected.

Conclusion

While dealing with the hardware, other than improving our theoretical knowledge, we learned about the practicality involved while creating an equipment and the enumerable factors that need to be considered when we do so. The joy of working in a team and the thrill involved while tackling the various problems and challenges faced to make an equipment solely by the knowledge we have gained so far gave us

a firsthand feel for what it's like be an engineer. We would like to extend our gratitude to Prof. Nandkumar Nambath for giving us such an opportunity and guiding us throughout the process whenever necessary.

References

- [1] Course Lecture Notes
- [2] Lab Course VHDL codes.