Mohamud Haroon

Project

Digital Clock

Declaration

I hereby declare that this is my original work produced without the help of any third party unless where referenced within this report.

Signed: Mohamud Haroon

Content Table

Abstract

This is a project report about building a digital clock. The digital clock will be able to have a voltage regulator from 9V to 5V and 3.3V, use IC chips to display numbers on the LCD, binary counter, LM555 timer, demultiplexer and the MSP430, will be instrumental to building this project. During the build of the digital clock, a good methodology throughout is to have all the surface mount devices on one side. This makes the job of using the oven much faster and simpler. It is also a good practice to us e silkscreen on the design phase to correctly label + and – connections and labelling IC positions. This also makes it easier to analyse the circuit and make connections clear thus making it a circuit that is a lot more pleasant to work with. A thorough and detailed data will be taken from the oven when in progress of soldering the surface mount components to the PCB. Throughout this report, more about how these components work will be discussed and explain in further details.

Objectives:

- Learn about how digital clocks work
- Keep a record of weekly based work (log book)
- Build and test a 5V output voltage regulator
- Build and test a 3.3V output voltage regulator
- Research the difference monostable and Astable multivibrator circuits for the 555 timer.
- Build and test an oscillator.
- Learn about LCD and how to use drivers and demultiplexers with it
- Learn about the 74HCT238 decoder.
- Write a code to test the hardware.
- Use switches and a buzzer.

- Design a PCB
- Learn about surface mount soldering
- Use soldering oven

Acknowledgements:

During the build and component researches for the project, I have received various supports in the form of individual to individual, class explanation, and website supports. I received support both Patrick O’Friel and Tom Murray. Class explanations were very supportive as they explain the how the components come together to work together. The websites are very important as they are the most frequently consulted source of information.

Introduction:

- **Voltage Regulator:**

The digital Clock project is capable of counting up to 24 hours. The DIP50 LCD is the display unit. The MSP430 is programmed to produce an output which is used as the time. Which is then displayed using the LCD. To decode the output of the MSP430 into segments for the LCD, the 74HCT4543 IC chips are used. One for each digit. The MSP430 is powered with 3.3V and the all the IC chips in the project are power with 5v. The LCD, however, uses a 64Hz for clearing the screen. This is produced from the 4020 IC chip in the oscillator. If the IC chips, the MSP430 or the display unit, are directed connected the 9V battery, all the IC chips will be damaged, thus two voltage regulators are required. One that has an output of 5v and one with an output of 3.3V. The 5v powers the IC chips and the 3.3V powers the MSP430 and the switches. The LM7805 is suitable for the 5V regulator and the LM317 is suitable for the 3.3V regulator. See fig 1.1. The MSP430 produces outputs in binary. Which then is connected to the 6 LCD drivers. The drivers are 74HCT4020. These decode a binary into a 7-segment display unit. With this set up in place, If the binary signal is sent from the MSP430, all digits will display the same number because the data bus is connected to all the drivers. Therefore, a selection method is required using the latch pin of the drivers, which selects which digit is to be changed. A suitable component is the 74HCT238 demultiplexer. The demultiplexer selects a driver and makes it transparent and the other drivers will be latched, meaning that one digit will be changed at a time. The selection of the drivers using the demultiplexer is controlled by the MSP430. See fig 1.2.

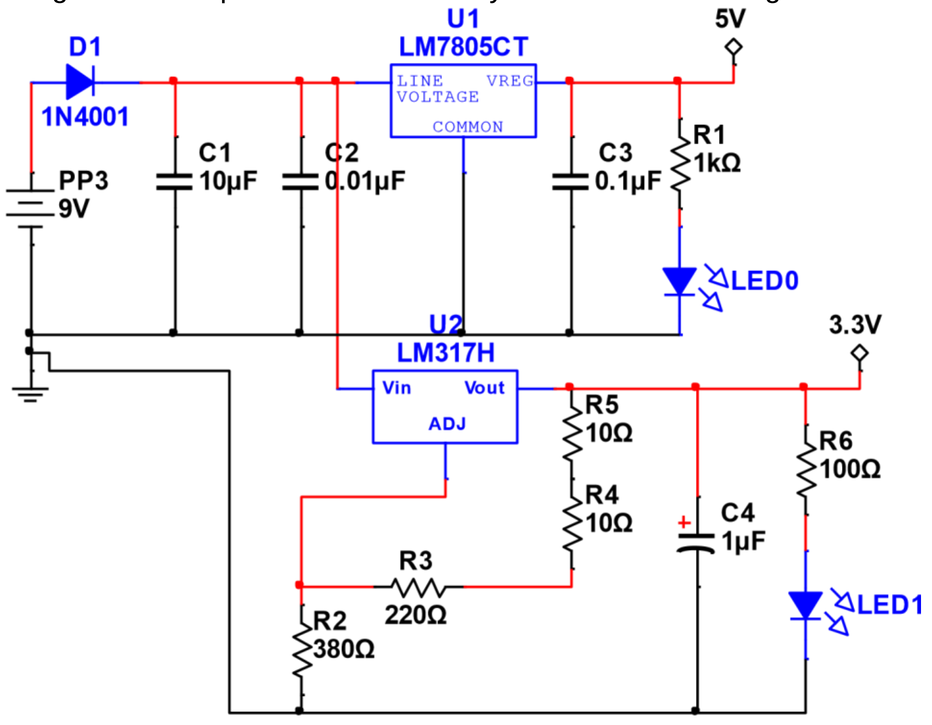


Fig 1.1

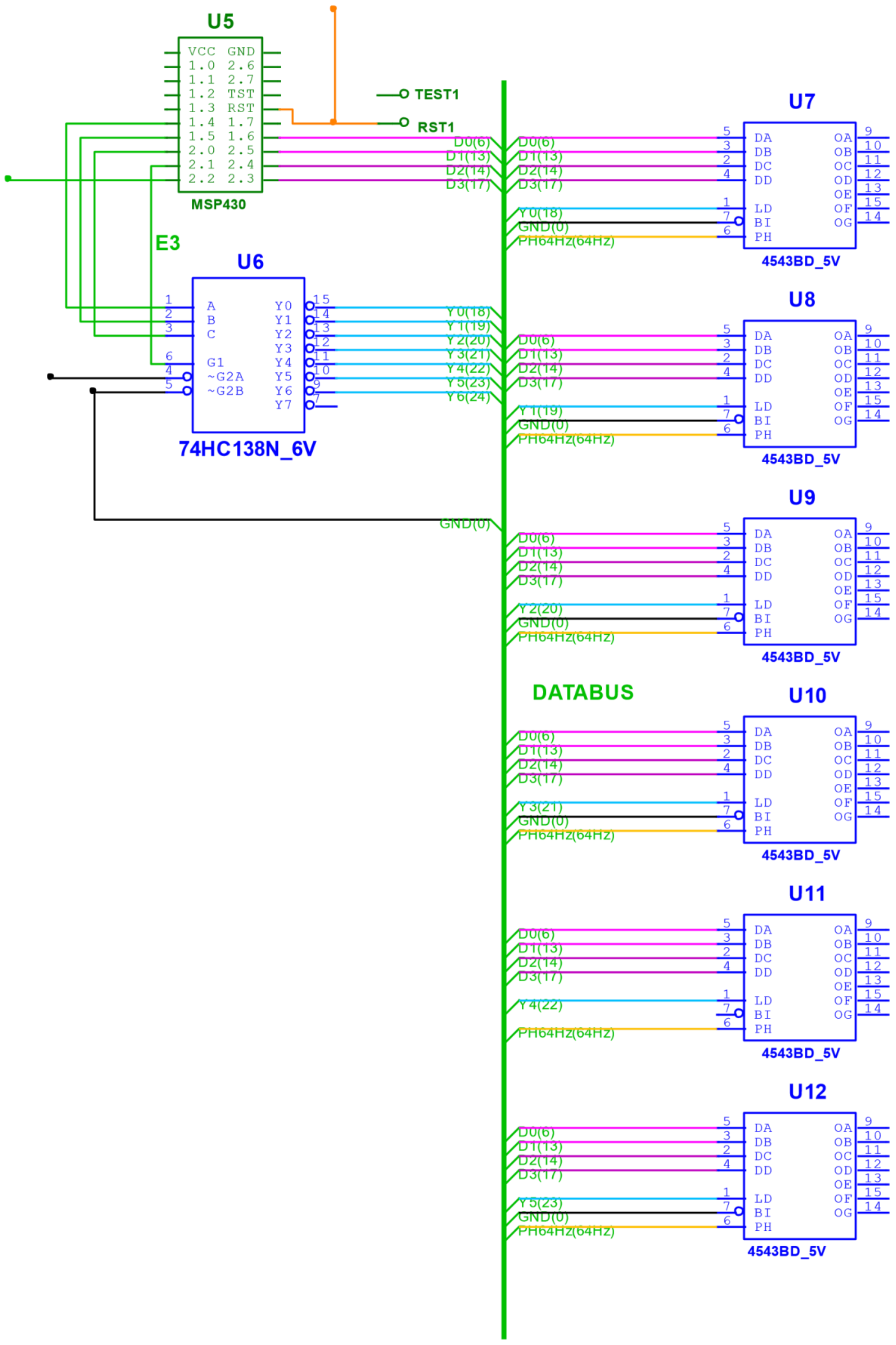

U5
MSP430
TEST1
RST1
E3
U6
74HC138N_6V
U7
U8
U9
U10
U11
U12
4543BD_5V
DATABUS
D0(6)
D1(13)
D2(14)
D3(17)
Y0(18)
Y1(19)
Y2(20)
Y3(21)
Y4(22)
Y5(23)
Y6(24)
GND(0)
PH64Hz(64Hz)

Fig 1.2

- **Oscillator:**

Once the voltage regulator has been built and tested, the next thing to do is the oscillator. The oscillator is a vital feature of the digital clock as it refreshes the LCD. The oscillator can have outputs of multiple frequencies at the same time. The LM555 timer, with the right schematic, generates a frequency of 8192Hz. This frequency is then sent out into the 14-stage binary counter, the 74HCT4020. See fig 1.3. The binary counter divides the frequency into different pins making its lowest frequency output to 0.5Hz. The frequency output from pin 6 of the 4020, which has an output frequency of 64Hz, is used to refresh the LCD screen. It is also connected to the Phase input pins of the drivers. See fig 1.2

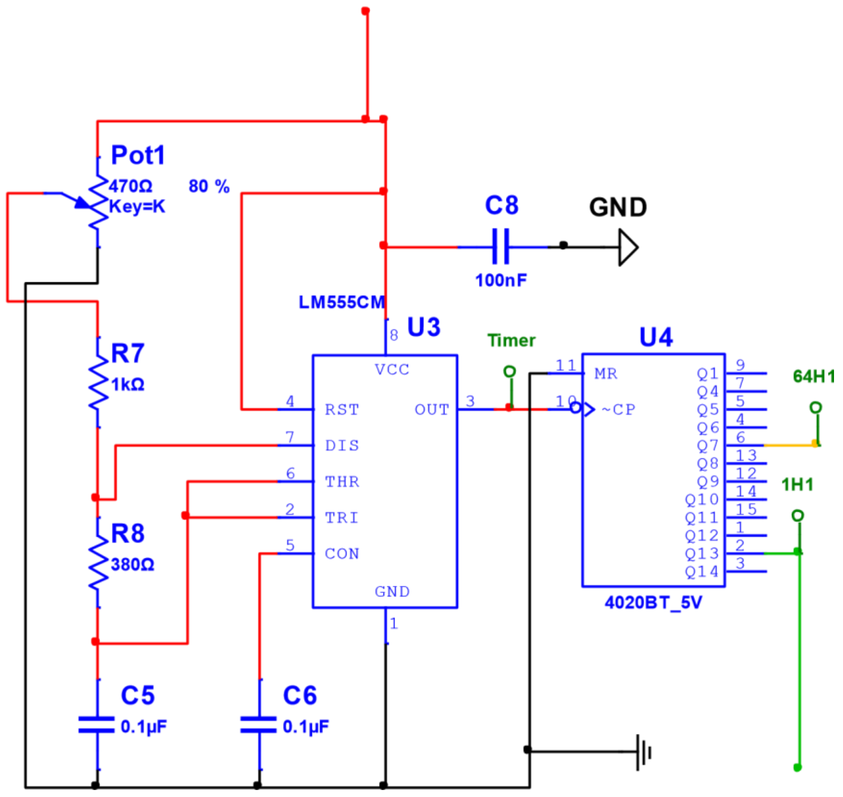
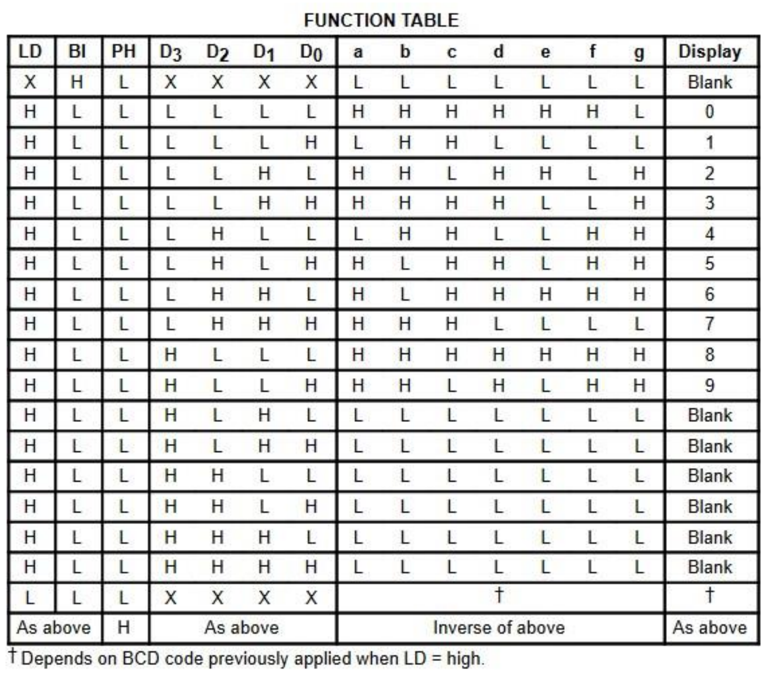


Fig 1.3

- **Display Unit**

The display unit is essential to the project. For prototype 1 of the project, one common anode 7-segment display and one common cathode 7-segment display are used as it on displays two digits. However, for prototype 2, a 6-digit LCD is the display unit. The LCD display unit has 50 pins. The drivers will be connected to the LCD's pins correctly as each pin can only be connected to the correct pin of the correct driver. Otherwise, the LCD will not function properly. This means that since the drivers are controlled by the MSP430, the display unit is also controlled by the MSP430. The MSP430 can send a binary signal to the drivers which will decode the signal to the display unit and will display the desired numbers. The display unit can only display Numbers only. This is because the drivers can only decode binary signals from 0-9. If the input binary is anything other than binary 0-9, the display unit will display blank. See fig 1.4 datasheet of the drivers. The LCD has a refresh rate of 64Hz, which is generated from the oscillator circuit. See Fig 1.3



FUNCTION TABLE

LD	BI	PH	D_3	D_2	D_1	D_0	a	b	c	d	e	f	g	Display
X	H	L	X	X	X	X	L	L	L	L	L	L	L	Blank
H	L	L	L	L	L	L	H	H	H	H	H	H	L	0
H	L	L	L	L	L	H	L	H	H	L	L	L	L	1
H	L	L	L	L	H	L	H	H	L	H	H	L	H	2
H	L	L	L	L	H	H	H	H	H	H	L	L	H	3
H	L	L	L	H	L	L	L	H	H	L	L	H	H	4
H	L	L	L	H	L	H	H	L	H	H	L	H	H	5
H	L	L	L	H	H	L	H	L	H	H	H	H	H	6
H	L	L	L	H	H	H	H	H	H	L	L	L	L	7
H	L	L	H	L	L	L	H	H	H	H	H	H	H	8
H	L	L	H	L	L	H	H	H	L	H	L	H	H	9
H	L	L	H	L	H	L	L	L	L	L	L	L	L	Blank
H	L	L	H	L	H	H	L	L	L	L	L	L	L	Blank
H	L	L	H	H	L	L	L	L	L	L	L	L	L	Blank
H	L	L	H	H	L	H	L	L	L	L	L	L	L	Blank
H	L	L	H	H	H	L	L	L	L	L	L	L	L	Blank
H	L	L	H	H	H	H	L	L	L	L	L	L	L	Blank
L	L	L	X	X	X	X				†				†
As above		H	As above				Inverse of above							As above

† Depends on BCD code previously applied when LD = high.

Fig 1.4

http://www.unicornelectronics.com/ftp/Data%20Sheets/74hct4543.pdf

- **MSP430**

The MSP430 is the brain of the project. This is where the software program will be stored in and the timing operation. The MSP430 manages the operation of digit selection on the display unit. The MSP430 also manages the switches and buzzers. When setting the alarm, the MSP430 memory is used to store the data. The data to be displayed on the LCD is sent from the MSP430 to the drivers which decodes the data and display it on the LCD. The MSP430 also manages this. The power it requires to be able to function and do all these tasks is 3.3V. this is pulled from the output of the 3.3V voltage regulator. See fig 1.5.

Project Planning

The project plan was to build of the project on Multisim, test each section on a patch board, build forward the Multisim file to Ultiboard and design the PCB to in a size no larger than 90 mm x 90 mm. Once the PCB arrives, solder surface mount components using oven. The most highlighted experience in this project was learning about the Surface mount Technology and it's soldering techniques. During the process of designing the PCB, some mistakes were made, which did put the project back in it's planned schedule. The finished design was mixed up with an unfinished design which was only missing connections. This took time to correct this with soldering and wires. Another mistake on my design was placing a buzzer in the PCB that has a 10mm difference. Such buzzer is not available in college. This also took time to be corrected. Furthermore, the reset switch of the MSP430 was wired incorrectly and also delayed the planned schedule.

Overview of the circuit:

The clock project is powered by a 9V PP3 Battery. The 9V is regulated to two values. The 5V voltage regulator powers all the IC chips except the MSP430 because the MSP430 is a low powered IC chip. The 3.3V voltage regulator powers the MSP430, buzzer and switches. The MSP430 sends a binary signal to the 74hct238 to select the desired driver. Then sends another binary signal to another signal which is the desired number, to the data bus that is connected to the drivers. This makes the selected digit change to the desired number on the LCD. The LCD has a refresh rate of 64Hz which is generated by the oscillator. See fig 1.5

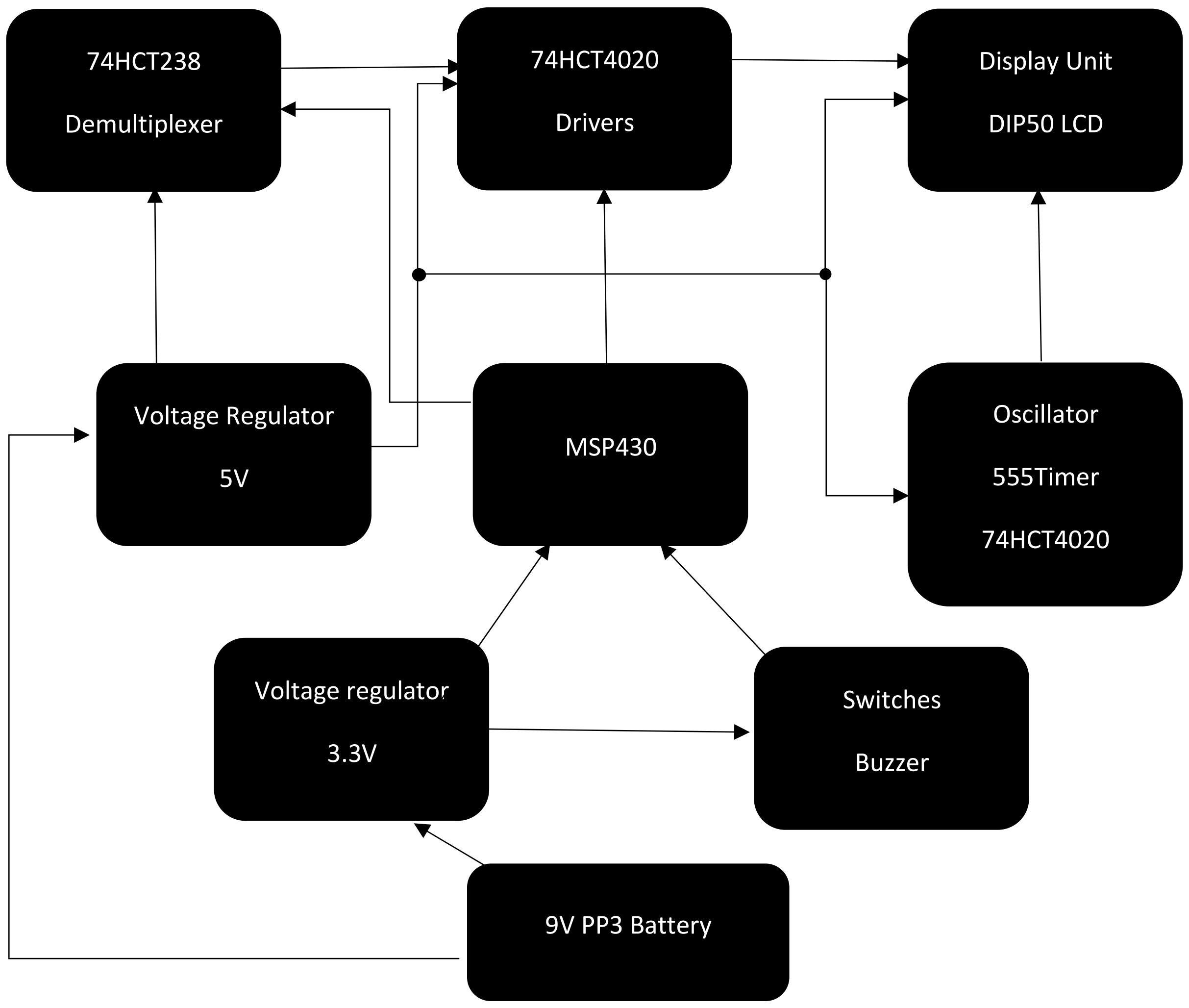


Fig 1.5

Theoretical operation of the circuit:

- **Voltage Regulator**

The 5V voltage regulator uses LM7805 which has an input voltage of 9V from the PP3 power supply battery going into pin 1 and an output voltage of 5V from pin 3. This is expanded throughout the circuit and is used by all the IC chips. The LM317 has 3 pins. Vin, Adjust and Vout. When making the 3.3V voltage regulator, pin 3 of the LM317 is connected to pin 1 of the LM7805. This means that there is an input voltage of 8.3V on the LM317 in pin 3. Adjust has ~5.3mA going through it. A resistor of value ~400Ω will be necessary to place between adjust and ground. This resistor value adjusts the output of the voltage regulator. This means that if adjust resistor is not ~400Ω, the output of the voltage regulator is not going to be 3.3V. A resistor with the value of 240Ω is also crossconnected to pin 1 from pin 2. This also helps voltage adjustment. See fig 1. The LM317 Vout formula is as follows

$$Vout = 1.25\left(1 + \frac{R_2}{R_3 + R_4 + R_5}\right) + IR_2$$

$$Vout = 1.25\left(1 + \frac{400}{240}\right) + 5.262 \times 10^{-3}$$

$$Vout = 3.33V$$

- **Oscillator**

The oscillator is the timer of the digital clock. The IC chips on the oscillator are the LM555 timer and 74HCT4020 which is the 14-stage binary counter. The LM555 timer sends a frequency of 8192Hz to the clock pin of the binary counter which has a duty cycle of 77.3% high. See 1.6. For the LM555 timer, there are two modes that can be built. A monostable and an Astable mode. There are minor differences between the two modes, however, Astable mode is the correct mode to use for this project. Once the timer has been constructed correctly, the output frequency on pin 3, 8192Hz is connected to the clock pin of the binary counter, pin 10. See fig 1.3. The pot is an adjustment to the frequency output of the LM555 timer. It is what determines the accuracy of the clock. Depending on the pot resolution and the resistors connected in series to the pot, R7 and R8, you can have as accurate of a clock as you intend. The binary counter divides frequency and then has an output pins of different frequencies. Pin 6 has output frequency of 64Hz which is used for the LCD and the drivers. See FigA1.

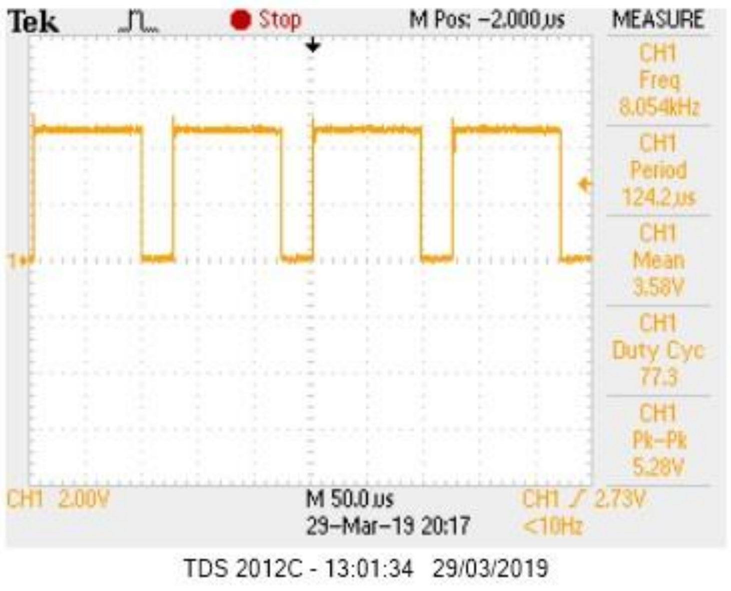


Fig 1.6

- **Display**

This section is the output section. It is important that it is visually well presented from a design's point of view. The LCD has 6 digits as it needs to keep track of numbers for 24 hours. This means that the 6 digits require a driver each. If the phase input of the decoder is turned low, pin 9 - 15 are turn active high and if phase input is turned high, pin 9 - 15 are turned active low. However, since LCD is used, the phase input pin is connected to the same frequency as the LCD backplane. For more information about the drivers, see 74hct4543 datasheet link in the reference. Binary input pin will be turned high for all of the drivers as it turns the display to blank. See fig 1.4 and fig-A4. The LD pin is turned high for transparent and turned low to latch. This means that when LD is high, any input from D0 – D3 will be decoded and displayed on the selected digit. When the LD is latched, it means that the last thing displayed before LD was turned low will be displayed even if D0-D3 are changed. LDs on all drivers are connected to the demultiplexer, the 74HCT238. The demultiplexer acts as a selector for which drivers to turn transparent and which one to turn latched. See fig 1.2. The LD pins of the drivers are connected to the 74HCT238 demultiplexer. Pin Y0 is connected to driver digit 1 and Y5 connected to driver digit 6. E3 is turned high as E1 and E2 are turned low. A0-A2 are connected to MSP430 and are t and so on. See e selectors of which Y to turn high. If the address which is A0-A2, binary 000, then Y0 is selected transparent, if address is 101, Y5 is transparent. See fig-A2 and fig-A3.

- **MSP430**

The MSP430 is what will bring all the sections together and make it all work together. It is powered with 3.3Vs from the LM317 voltage regulator. The software will be uploaded to this chip which manages the timing operation and the display of numbers through sending binary signals to the drivers and the demultiplexer. The software that is uploaded to the MSP430 chip uses the millis() function. The millis function has a limit of 50 days before it resets back to zero. See millis() function link in reference. See fig-A5 for the completed code.

- **Soldering**

When soldering using the oven, the solder mask and the components must be properly placed on the PCB.

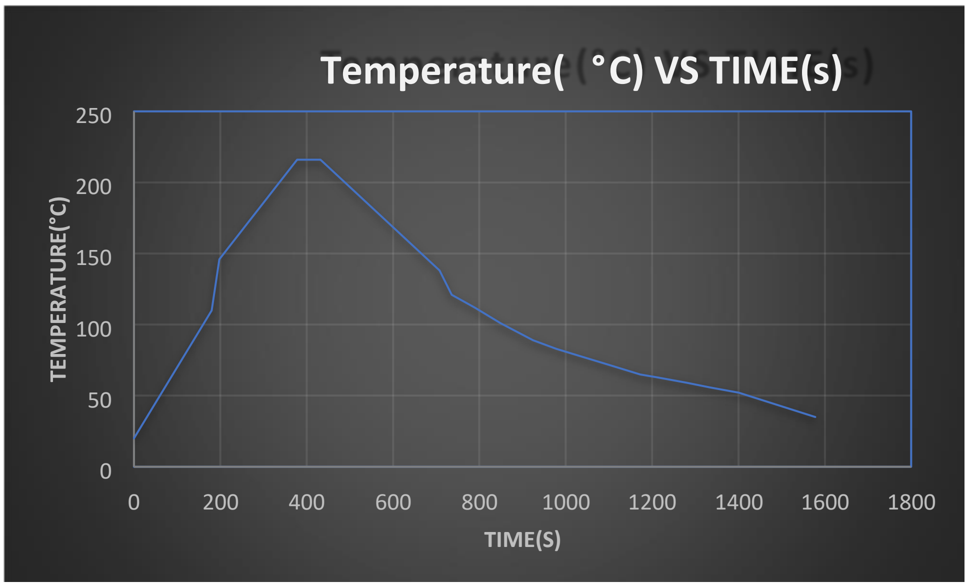
Safety

- Oven must be properly connected to the main power supply and properly grounded
- Only operate on dry condition to prevent electric shock
- Do not operate the oven in an environment which has flammable gasses, dust or steam is present.
- Only use oven for PCBs and nothing else

Once the PCB is placed on the oven, the LED's indicate the state of the oven and its phases

- **Preheat**: Slowly warms up and keeps the board heated for a time.
- **Soaking**: Flux activates, Oven continues to heat slowly to just under the melting point of the flux.
-
- **Reflow**: Heat increases quickly up to the melting point of the flux
- **Dwell**: Temperature is maintained slightly above the flux melting point

The process takes about 25 minutes. See The graph of the temperature vs time below.



Testing:

The testing is carried out when the PCB arrives and notes are taken down on the log book of any incorrections observed and tested. The tests have caught some mistakes from the design phase. All found faults are documented on a log book. There were 15 incomplete connections on the PCB once it arrived. The connects took time to complete but they were all resolved. After all the section have been soldered on the PCB. Sockets have been used for All of the IC chips and the LCD. This to prevent any damage to the MSP430 and in case a problem arise that might have been overlooked. While testing the board, I have noticed that the binary inputs of the drivers were connected to ground where as they were meant to be connected to 5v for this specific LCD. All the binary input tracks were connected and only 1 track was connected to ground. this was an easy solution which was to cut that track and connected it to 5V.

Summery:

Doing this project, I have learned

- How to build a 5V voltage regulator
- How to build a 3.3V voltage regulator
- About the LM555 timer and its modes
- The 4020 binary counter
- How to use LCD with drivers and demultiplexers
- How to use the millis() function • About surface mount technology.
- How to use the oven
- Design PCB
- About Interrupts

The digital Clock did not work as intended due to some problem with the MSP430. The code is not working either. I must get a new MSP430 chip and board.

Conclusion:

In conclusion I have achieved my goal of learning about how digital clocks work. I have also achieved meeting the deadline and handing up the project on the due date. I have improved my research and analysing skills. I have also improved in soldering and planning ahead in building on a circuit board. I have learned to design and debug a PCB. I have learned about surface mount technology and how to solder it on boards. I have learned that project planning does not always meet your expectation. If I had another chance of doing this project again, I would be very care full of mixing up the PCB files. I would also not use a DIP50 LCD and would rather use a 16 pin LCD with a blacklight. This LCD is much simpler, faster and easier to work with. Setting it up is software based which makes it much more attractive in my opinion.

References:

Bill of Material does not fit in an A4 page so it will be in the same folder as the report As well as the code.

74GCT4543 datasheet

http://www.unicornelectronics.com/ftp/Data%20Sheets/74hct4543.pdf Demultiplexer

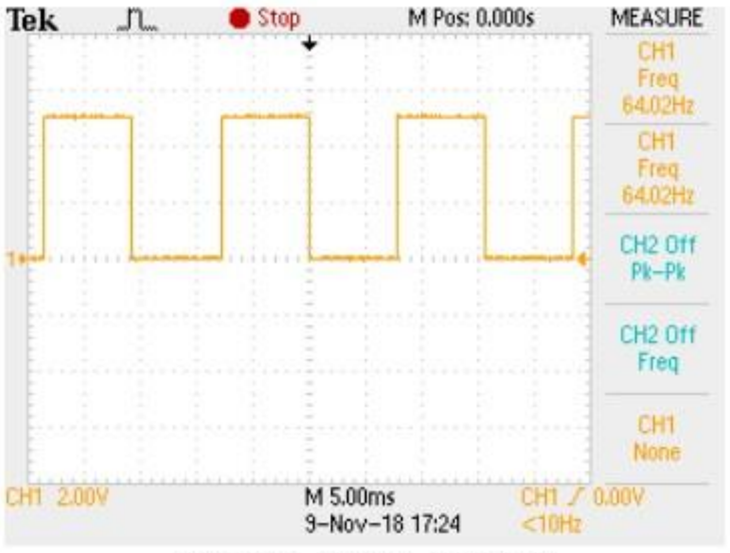
datasheet

http://www.ti.com/lit/ds/symlink/cd74hc138.pdf millis()

Function

http://energia.nu/reference/en/language/functions/time/millis/

Appendix:



TDS 2012C - 10:06:27 09/11/2018

Fig-A1 – Pin 6 Binary Counter 64Hz output

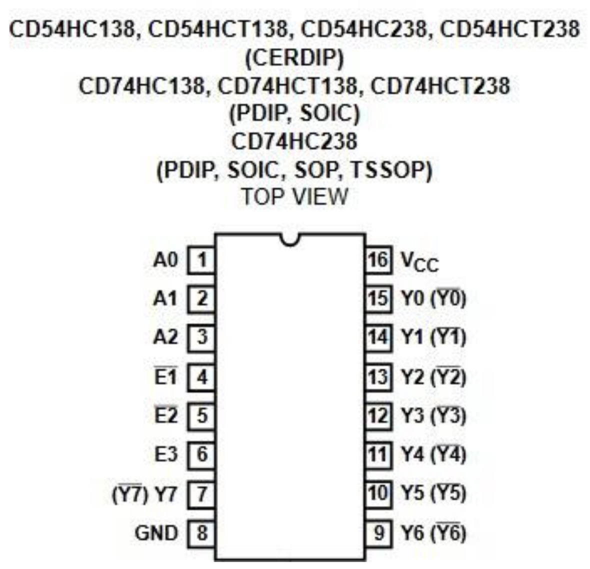
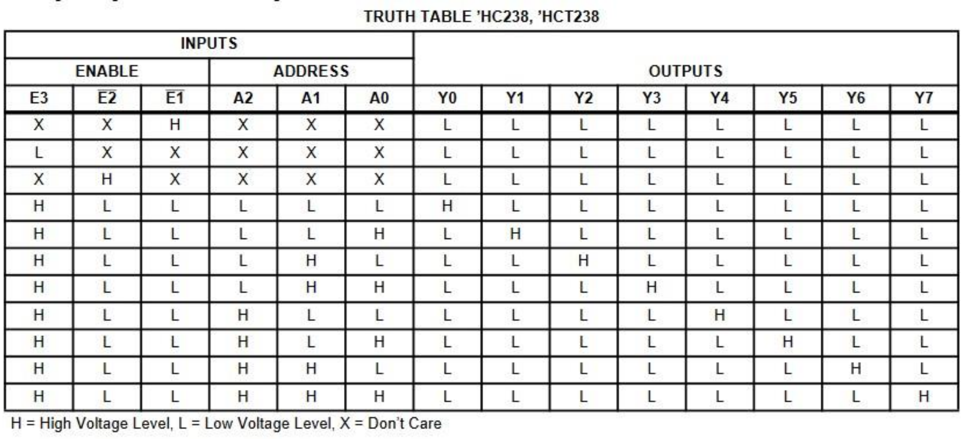


Fig-A2 – Demultiplexer Pins

http://www.ti.com/lit/ds/symlink/cd74hc138.pdf



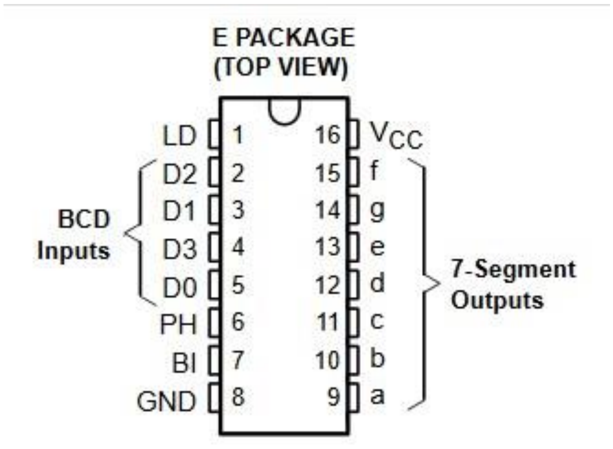
TRUTH TABLE 'HC238, 'HCT238

INPUTS						OUTPUTS							
ENABLE			ADDRESS										
E3	$\overline{E2}$	$\overline{E1}$	A2	A1	A0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	X	H	X	X	X	L	L	L	L	L	L	L	L
L	X	X	X	X	X	L	L	L	L	L	L	L	L
X	H	X	X	X	X	L	L	L	L	L	L	L	L
H	L	L	L	L	L	H	L	L	L	L	L	L	L
H	L	L	L	L	H	L	H	L	L	L	L	L	L
H	L	L	L	H	L	L	L	H	L	L	L	L	L
H	L	L	L	H	H	L	L	L	H	L	L	L	L
H	L	L	H	L	L	L	L	L	L	H	L	L	L
H	L	L	H	L	H	L	L	L	L	L	H	L	L
H	L	L	H	H	L	L	L	L	L	L	L	H	L
H	L	L	H	H	H	L	L	L	L	L	L	L	H

H = High Voltage Level, L = Low Voltage Level, X = Don't Care

Fig-A3 – Demultiplexer Truth Table

http://www.ti.com/lit/ds/symlink/cd74hc138.pdf



FigA8 – 7-segment decoder

http://www.unicornelectronics.com/ftp/Data%20Sheets/74hct4543.pdf

Digital Clock

```
//Name:      Mohamud Haroon
//No.:       X00145689
//Revision:  1.1
//Title:   digital clock code
//Description:    This code uses millis() function to count up to 24 hours
and uses the
//pins to output the clock to drivers and demultiplexers. the output is to
a 6 digit LCD
/*
   The Circuit

============================================================================
============================================================================
=====
   MSP pins                  Connected to              Comment
   ---------------------------------------------------------------
   01 - VCC                    3.3V                      Power Input pins
02   P1_0                  Buzzer                    For audio output if
desired
03 - P1_1                  LED                       Flashes when Alarm is
   on
04 - P1_2                  ---                       Not Connected
05 - P1_3                  ---                       Not Connected
06 - P1_4                  A0                        Connected to
74HCT238 Pin 1 for address selection
07 - P1_5                  A1                        Connected to 74HCT238
   Pin 2 for address selection
08 - P2_0                  A2                        Connected to 74HCT238
   Pin 3 for address selection
09 - P2_1                  E3                        Connected to 74HCT238
   Pin 6. Used to for Latch displayed numbers on the LCD when pulled low
10 - P2_2                  1Hz                       Output from
74HCT4020 Pin 2. Used for incrementing seconds. (Not used for this program)
11 - P2_3                  D3                        Data input Pin for the
LCD
12    - P2_4                  D2                        Data input Pin for
the LCD
13    - P2_5                  D1                        Data input Pin for
the LCD
14    - P1_6                  D0                        Data input Pin for
the LCD
15    - P1_7                  ---                       Not Connected
16 - RST                   RST                       Resets the MSP is
turned LOW
17 - TST                   TST
18 - P2_7                  ---                       Not Connected
19 - P2_6                  ---                       Not Connected
20 - GND                   GND                       GND pin on the MSP


============================================================================
============================================================================
=====
*/
#define Addr0 P1_4
#define Addr1 P1_5
```

Digital Clock

```
#define Addr2 P2_0

#define E3 P2_1

#define D0 P1_6
#define D1 P2_5
#define D2 P2_4
#define D3 P2_3
 int DG1; int
DG2; int DG3; int
DG4; int DG5; int
DG6;  int sec =
0; int sLSD = 0;
int sMSD = 0; int
mins = 0; int
mLSD = 0; int
mMSD = 0; int hrs
= 0; int hLSD =
0; int hMSD = 0;
int lastTime = 0;
 void setup() {
Serial.begin(9600);
pinMode(D0, OUTPUT);
pinMode(D1, OUTPUT);
pinMode(D2, OUTPUT);
pinMode(D3, OUTPUT);
   pinMode(Addr0,
OUTPUT);   pinMode(Addr1,
OUTPUT);   pinMode(Addr2,
OUTPUT);

  pinMode(E3, OUTPUT);

}  void
loop() {
  int timeNow = millis();   //Counting milli seconds

  if ((timeNow - lastTime) >= 1000) {
    lastTime = timeNow;       //if timeNow is 1000, raise a flag and
execute the following code.
    if (sec >= 59) {
sec = 0;
      mins++;                // if seconds is >= 59, increment min and reset
seconds
      if (mins >= 59) {
mins = 0;
        hrs++;              // if minutes is >= 59, increment hrs and reset
mins
        if (hrs >= 23) {           sec = 0;           mins = 0;
hrs = 0;           // if hours >= 23, reset all digits.
          //timeNow = 0;
        }
      }     }
else sec++;
printTime();
```

```
  }    sLSD = sec %
10;   sMSD = sec /
10;   mLSD = mins %
10;   mMSD = mins /
10;   hLSD = hrs %
10;
  hMSD = hrs / 10;        //Converting digit of hrs mins and seconds to 6
digits


  DG6 = sLSD;
  DG5 = sMSD;
  DG4 = mLSD;
  DG3 = mMSD;
  DG2 = hLSD;
  DG1 = hMSD;         // assign digits to hours, minutes and seconds.
   digitalWrite(D0, bitRead(DG6,
0));   digitalWrite(D1, bitRead(DG6,
1));   digitalWrite(D2, bitRead(DG6,
2));
  digitalWrite(D3, bitRead(DG6, 3));        //assign significance of each
bit
  digitalWrite(E3, HIGH);
  digitalWrite(Addr0, HIGH);              //1
digitalWrite(Addr1, LOW);              //0         Y5
digitalWrite(Addr2, HIGH);             //1   digitalWrite(E3,
LOW);
  delay(50);                              // select digit 6 as transparent
   digitalWrite(D0, bitRead(DG5,
0));   digitalWrite(D1, bitRead(DG5,
1));   digitalWrite(D2, bitRead(DG5,
2));
  digitalWrite(D3, bitRead(DG5, 3));    //assign significance of each bit

  digitalWrite(E3, HIGH);
  digitalWrite(Addr0, LOW);           //0
digitalWrite(Addr1, LOW);           //0           Y4
digitalWrite(Addr2, HIGH);          //1   digitalWrite(E3,
LOW);
  delay(50);                    // select digit 5 as transparent
   digitalWrite(D0, bitRead(DG4,
0));   digitalWrite(D1, bitRead(DG4,
1));   digitalWrite(D2, bitRead(DG4,
2));
  digitalWrite(D3, bitRead(DG4, 3));    //assign significance of each bit

  digitalWrite(E3, HIGH);
  digitalWrite(Addr0, HIGH);          //1
digitalWrite(Addr1, HIGH);          //1           Y3
digitalWrite(Addr2, LOW);           //0   digitalWrite(E3,
LOW);
  delay(50);                            // select digit 4 as transparent
   digitalWrite(D0, bitRead(DG3,
0));   digitalWrite(D1, bitRead(DG3,
1));   digitalWrite(D2, bitRead(DG3,
2));
```

```
  digitalWrite(D3, bitRead(DG3, 3));    //assign significance of each bit

  digitalWrite(E3, HIGH);
  digitalWrite(Addr0, LOW);             //0
digitalWrite(Addr1, HIGH);            //1        Y2
digitalWrite(Addr2, LOW);             //0   digitalWrite(E3,
LOW);
  delay(50);                           // select digit 3 as transparent
   digitalWrite(D0, bitRead(DG2,
0));   digitalWrite(D1, bitRead(DG2,
1));   digitalWrite(D2, bitRead(DG2,
2));
  digitalWrite(D3, bitRead(DG2, 3));    //assign significance of each bit

  digitalWrite(E3, HIGH);
  digitalWrite(Addr0, HIGH);           //1
digitalWrite(Addr1, LOW);            //0          Y1
digitalWrite(Addr2, LOW);            //0   digitalWrite(E3,
LOW);
  delay(50);                            // select digit 2 as transparent
   digitalWrite(D0, bitRead(DG1,
0));   digitalWrite(D1, bitRead(DG1,
1));   digitalWrite(D2, bitRead(DG1,
2));
  digitalWrite(D3, bitRead(DG1, 3));    //assign significance of each bit

  digitalWrite(E3, HIGH);
  digitalWrite(Addr0, LOW);              //0
digitalWrite(Addr1, LOW);              //0         Y0
digitalWrite(Addr2, LOW);              //0   digitalWrite(E3,
LOW);
  delay(50);                           // select digit 1 as transparent
}
```

Fig-A5