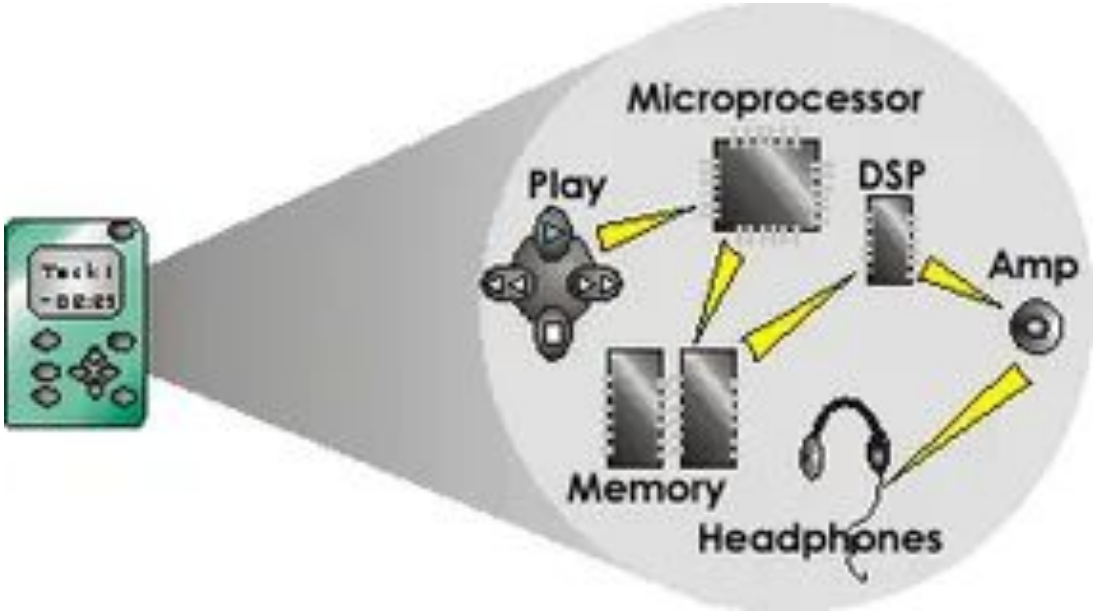
Office and Multimedia Applications

INTRODUCTION

Mp3 player

Portable music players have existed for decades, with the first truly portable consumer device released by Sony in 1979. Known as the “Walkman”, this cassette player for the first time allowed people to easily carry hours of music on their persons. Since then, technology in this field abruptly moved away from analog storage mediums to strictly digital ones. The CD player was the next revolution, as it was the first foray into mainstream digital music mediums. With the trend of shrinking the sizes of consumer electronics, and the growing popularity of sharing music online, the CD player was given up for flash-based music players. Dubbed the MP3 player, these devices allowed users far more convenience in customizing which music they wanted to load onto their players. Since flash memory was (and to some degree, still is) fairly expensive in terms of capacity per monetary cost, a shift to mechanical hard-disk drive based players began.

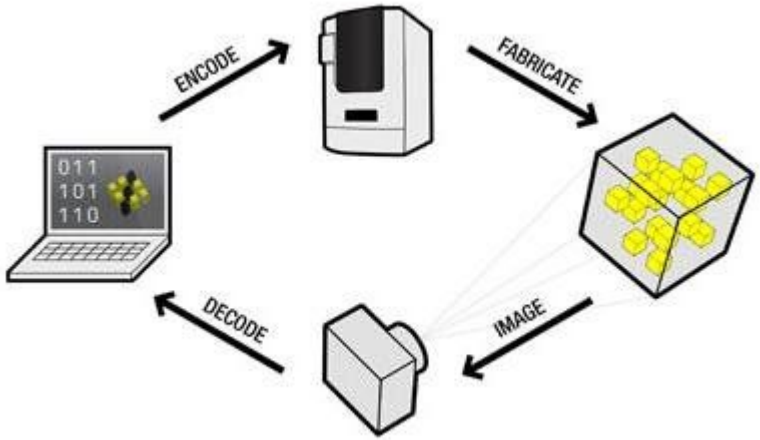
MP3 stands for MPEG Audio Layer III and it is a standard for audio compression that makes any music file smaller with little or no loss of sound quality. MP3 is part of MPEG, an acronym for Motion Pictures Expert Group, a family of standards for displaying video and audio using lossy compression.



3D Printers

Today the world we live in is becoming hugely dependent on technology. This means there is a need for a more technically skilled workforce to build and maintain required technology. Many new technologies are interactive, therefore it makes it easier to create environments in which learning can be done by doing, receiving feedback and refining understanding and building new knowledge. Arduino is an open-source platform used for constructing and programming of electronics.
It can receive and send information to most devices, and even through the internet to command the specific electronic device.

In short, 3D printers use CAD to create 3D objects from a variety of materials, like molten plastic or powders. 3D printers can come in a variety of shapes and sizes ranging from equipment that can fit on a desk to large construction models used in the making of 3D-printed houses.



Conceptual view of 3D printers and Mp3 players

3D Printers

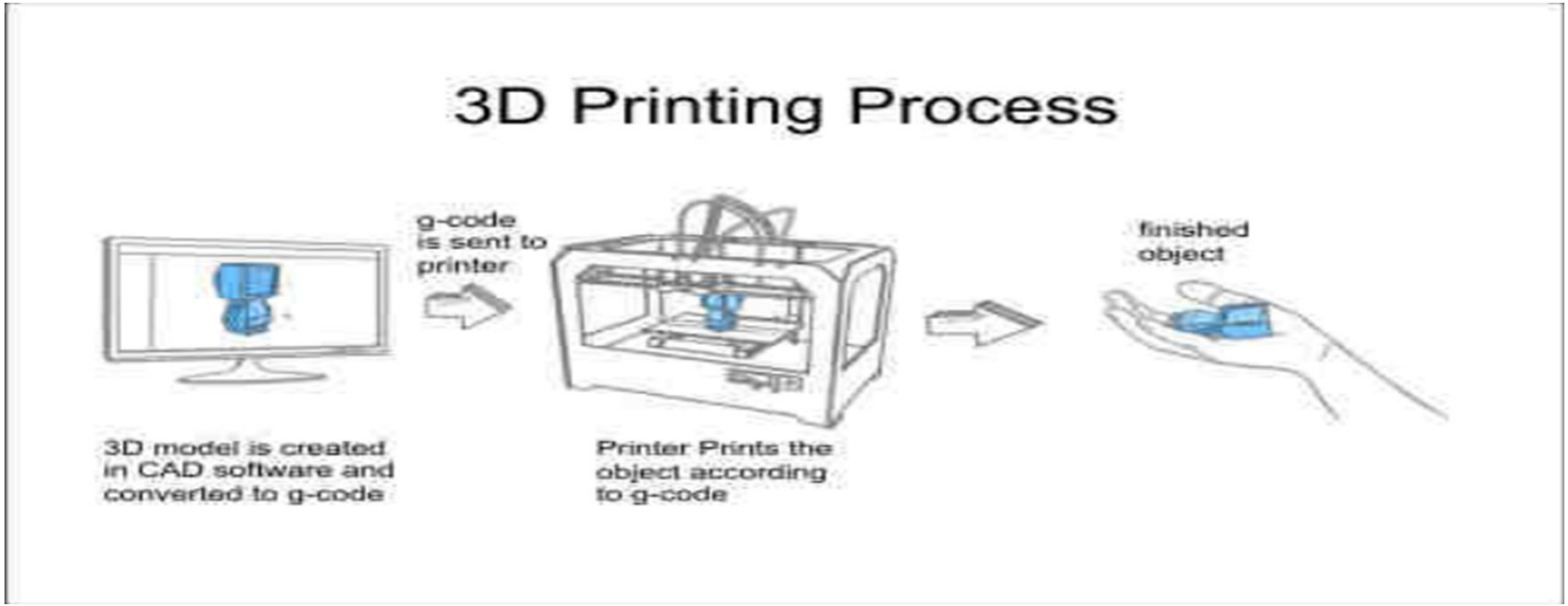
The 3D printer system design involves several complex components that work together seamlessly to produce a physical object from a digital model. The design of each component is critical to achieving high-quality prints and optimizing the printing process.

Mp3 players

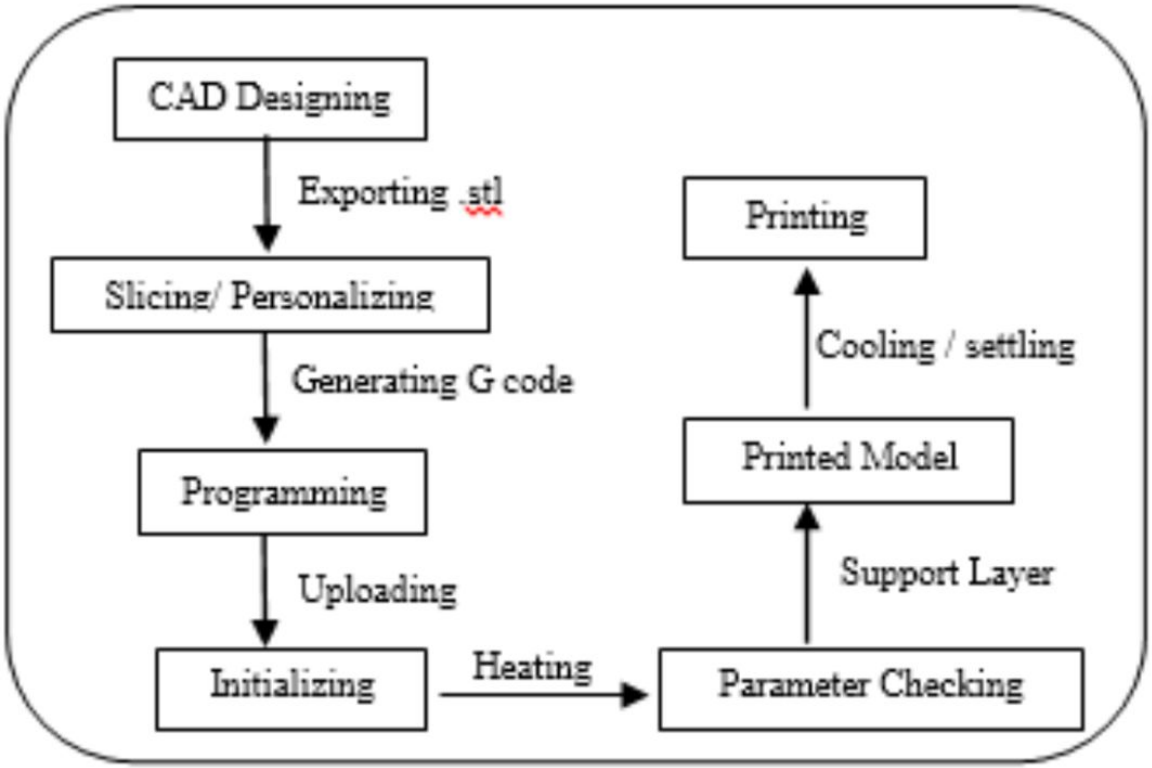
the system design of an MP3 player involves a combination of hardware and software components that work together to provide a seamless and enjoyable listening experience for users.

Microcontrollers are commonly used in MP3 players as they provide the necessary processing power and control to manage the various components of the device.

As shown in the fig. the proposed system can be divided into three modules.
They are
1. Actual Hardware System.
2. Interface System.
3. Computer Based Application.

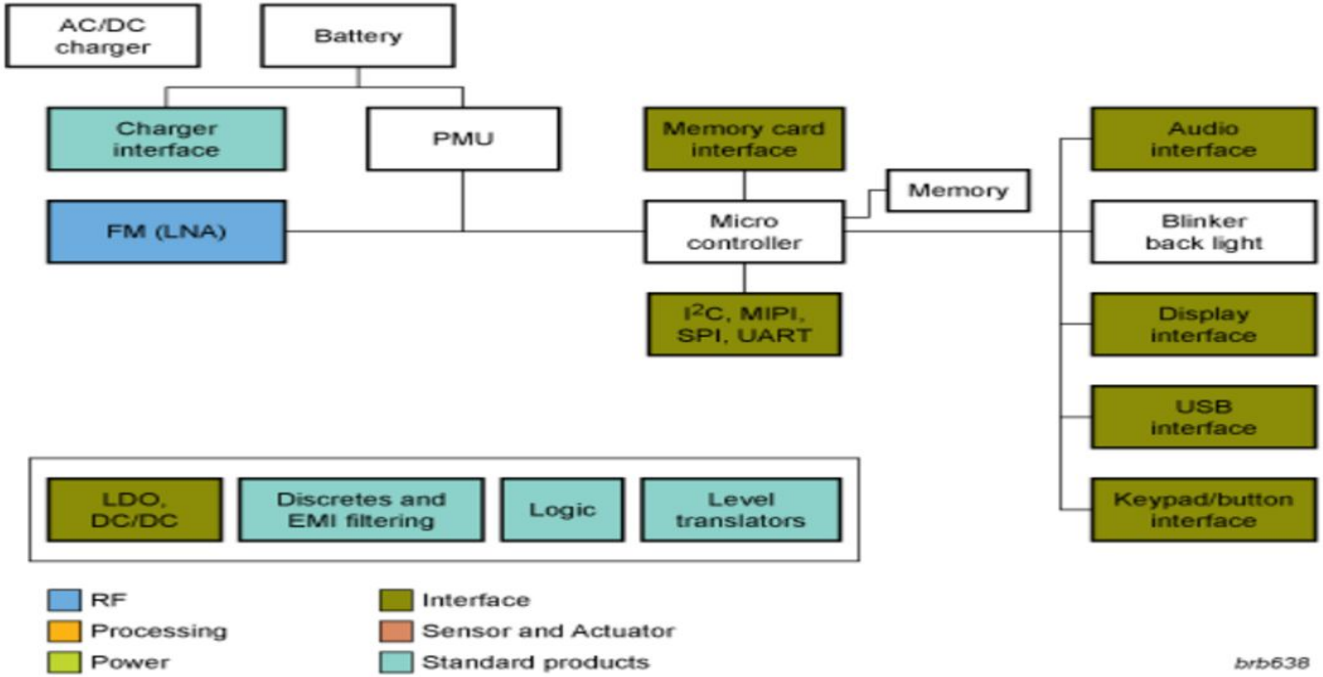


Block Diagram



Proposed system design for Mp3 Player

Block Diagram



Working and Features of 3D-Printer & MP3 Player

3D-Printer

A 3D-printer is a device that can create three-dimentional objects by adding successive layers of mater on top of one another.

Steps involved

1.Desgining the 3D model

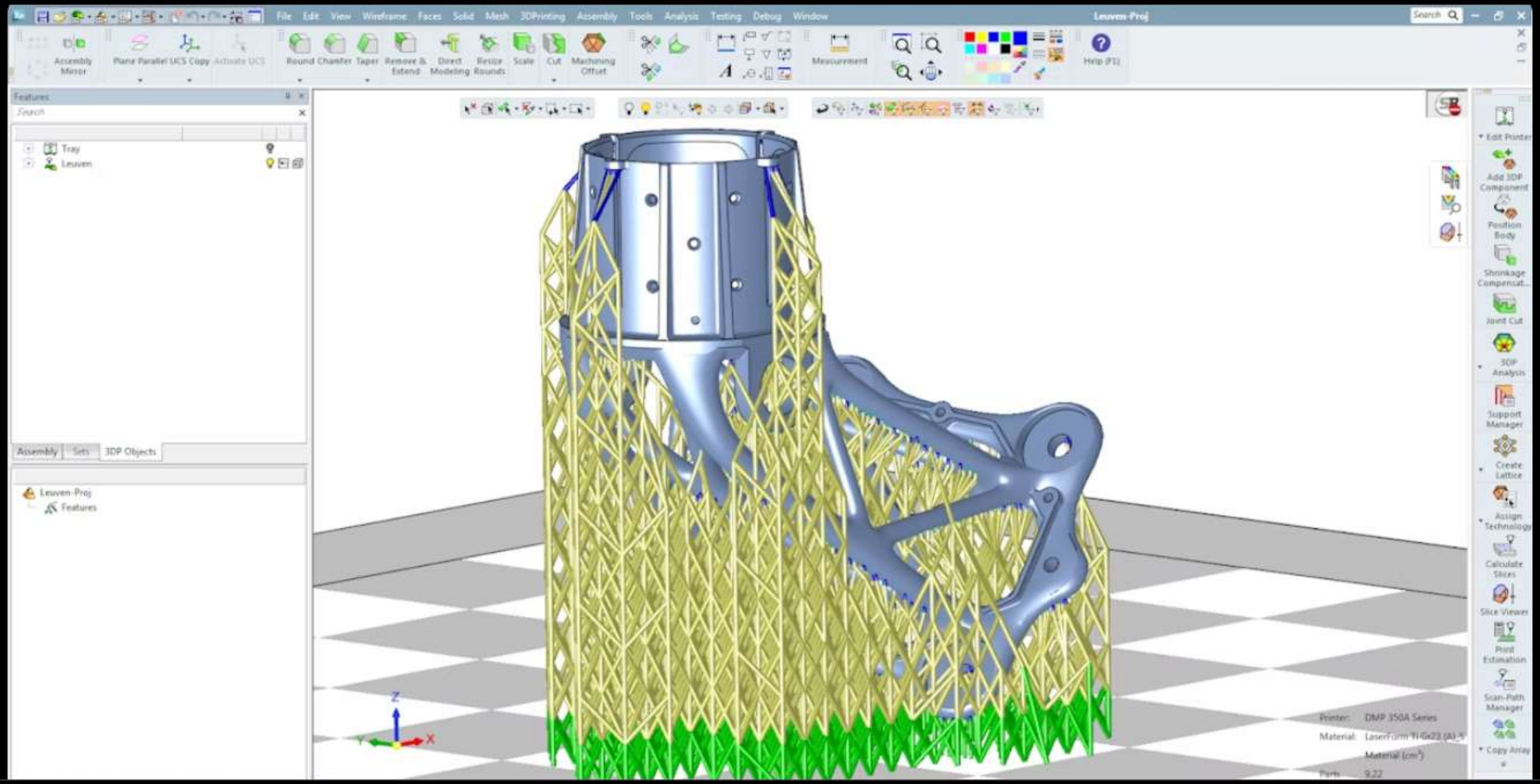
2.Slicing the model

3.Preparing the printer

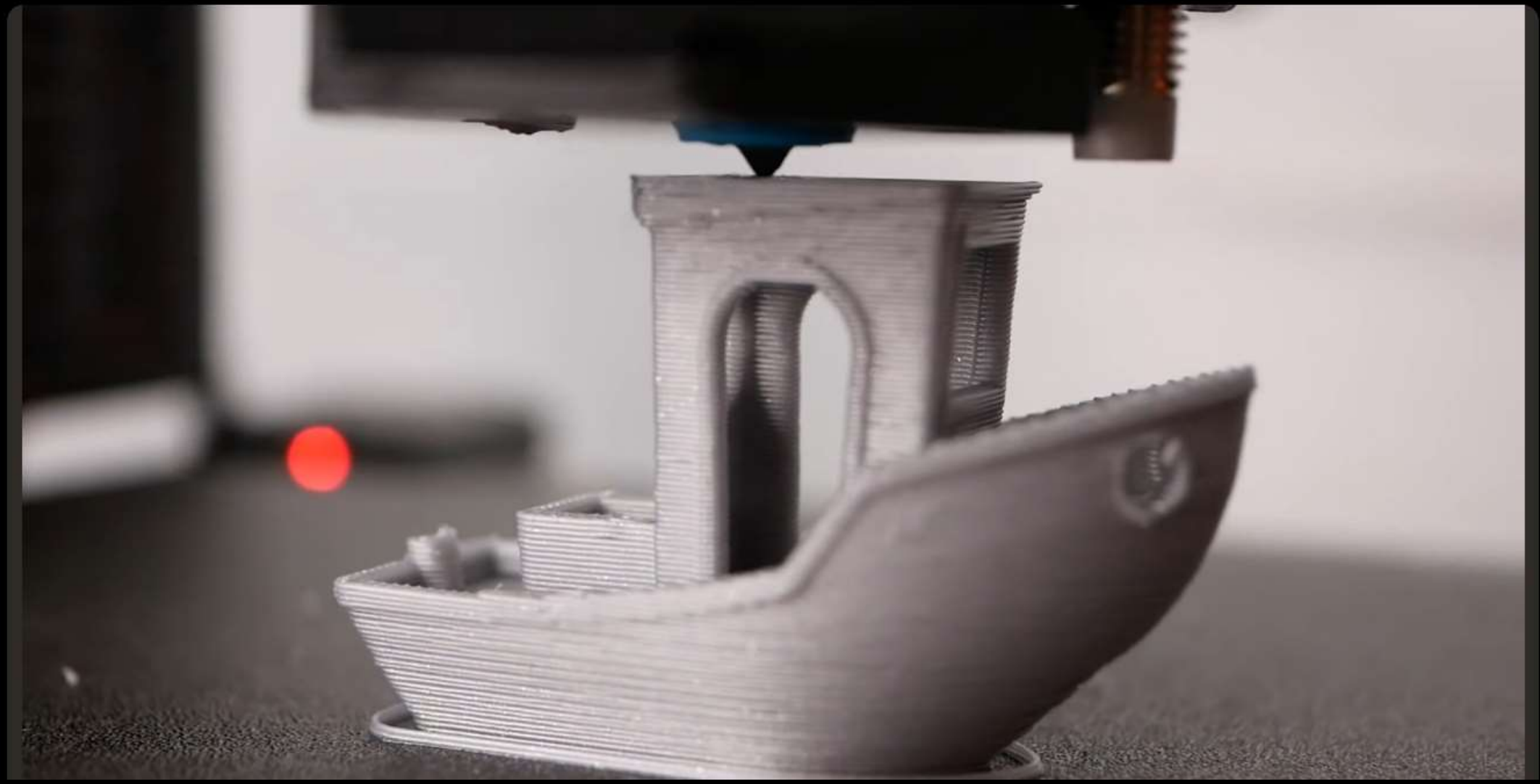
4.Printing the object

5.Finishing the object

DESGING THE 3D MODEL



SLICING THE MODEL

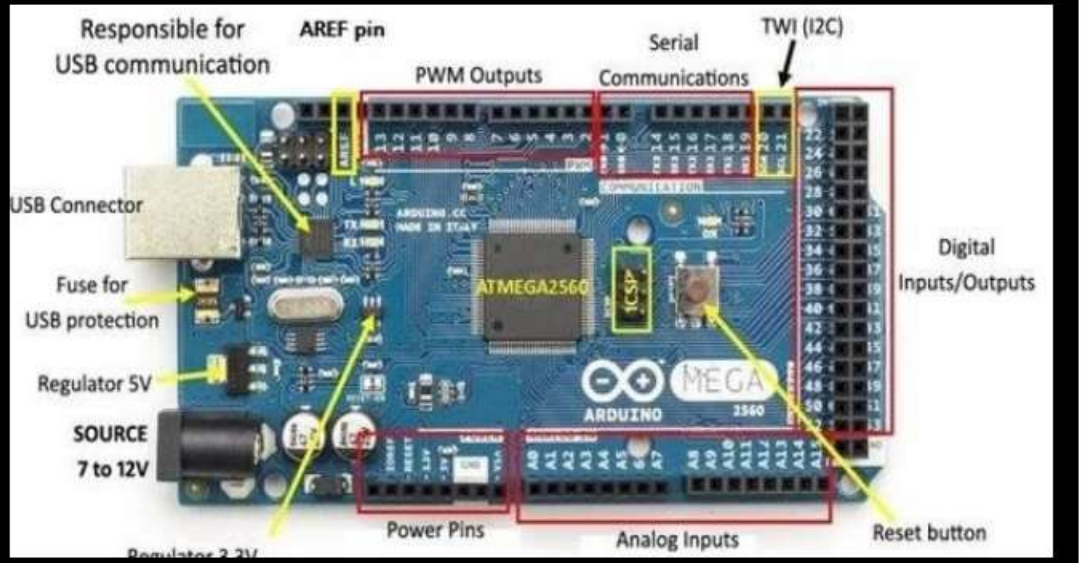


PREPARING THE PRINTER

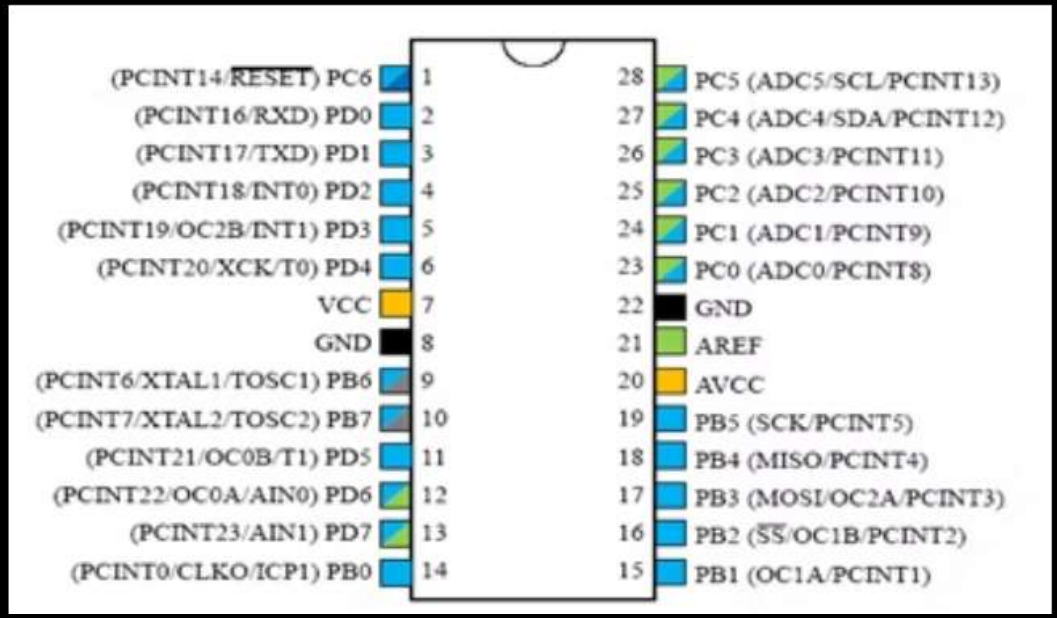
FINISHING THE OBBJECT

MICRO CONTROLLERS USED IN 3D PRINTER

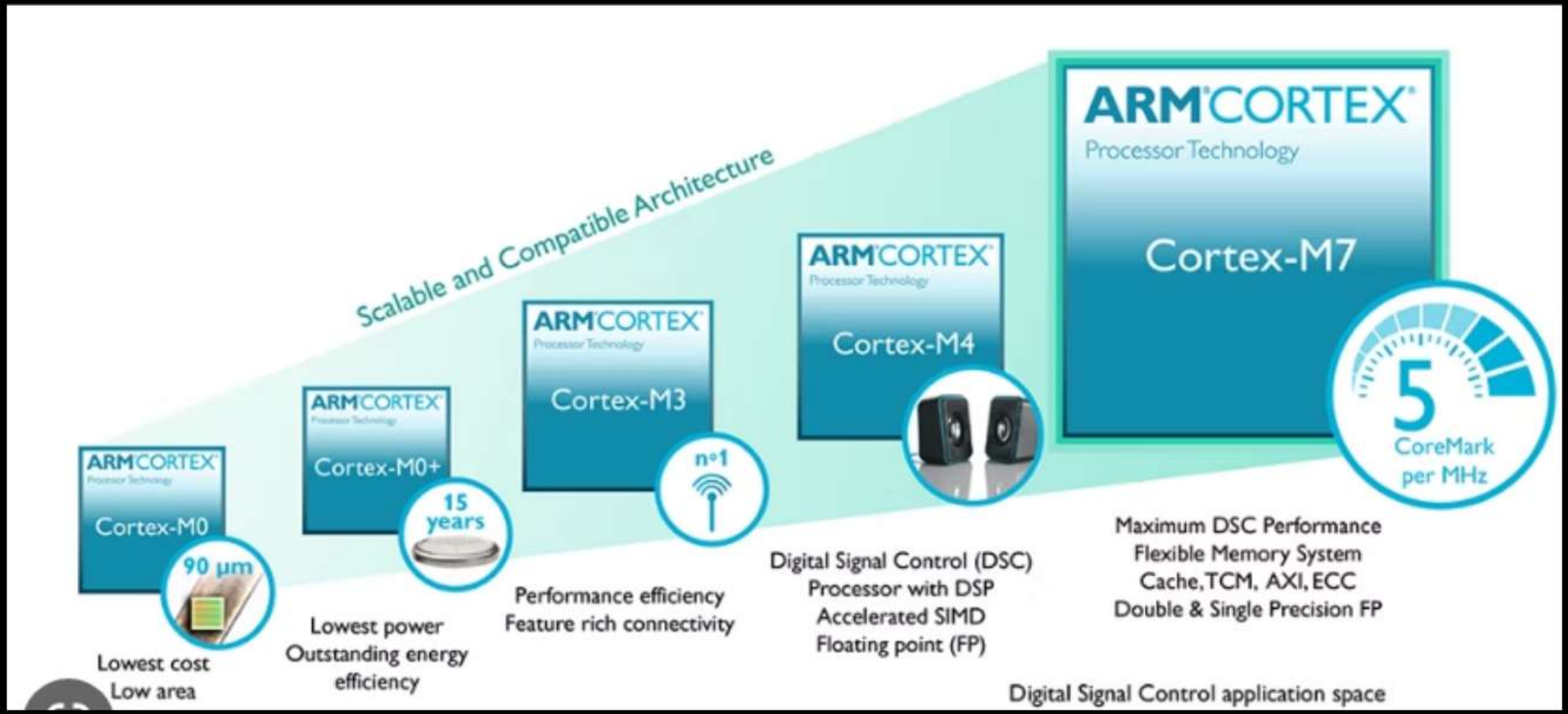
ATmega2560



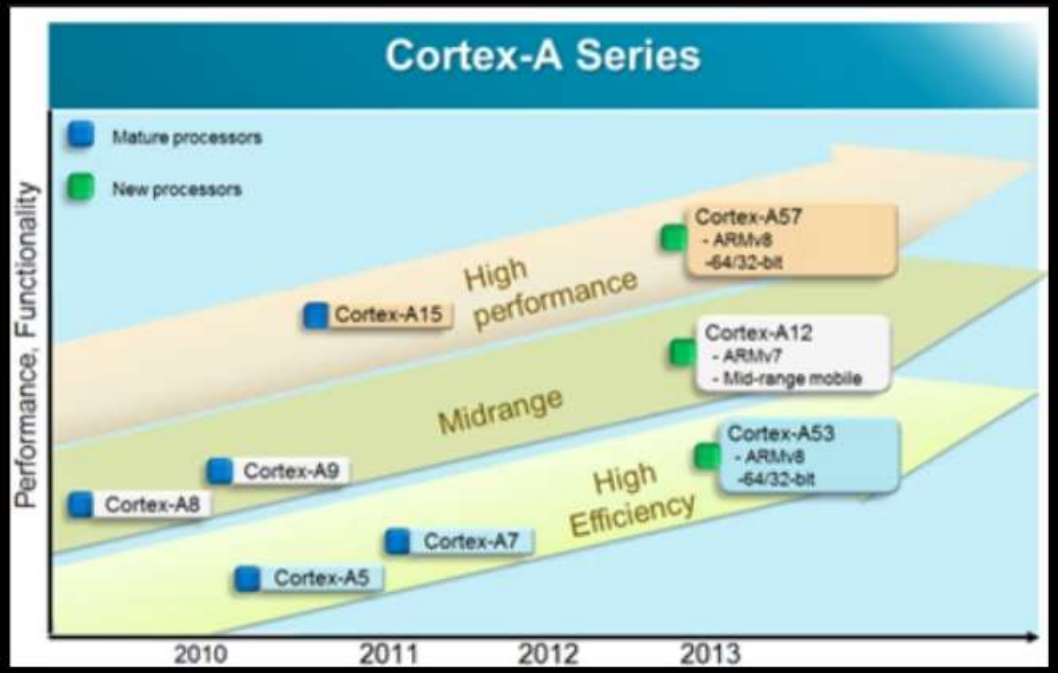
ATmega328P



ARM Cortex M-series



MICROPROCESSORS USED IN 3D-PRINTER



MP3 PLAYER

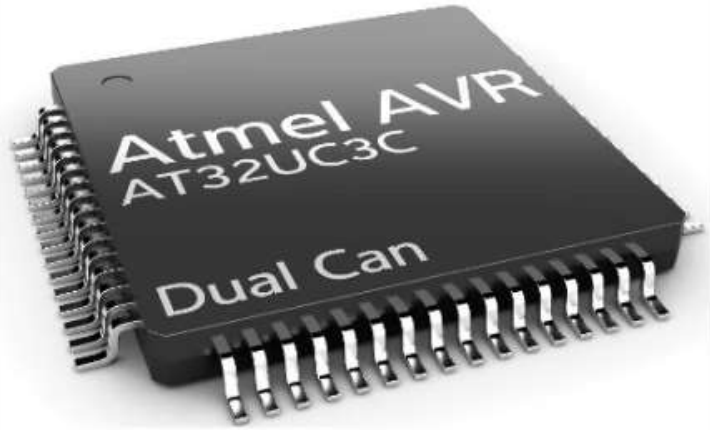
MP3 Player is a portable digital audio player that can store and play digital audio files in mp3 file format and this format compresses audio files size and makes to store thousands of files in small device

Four major componenets

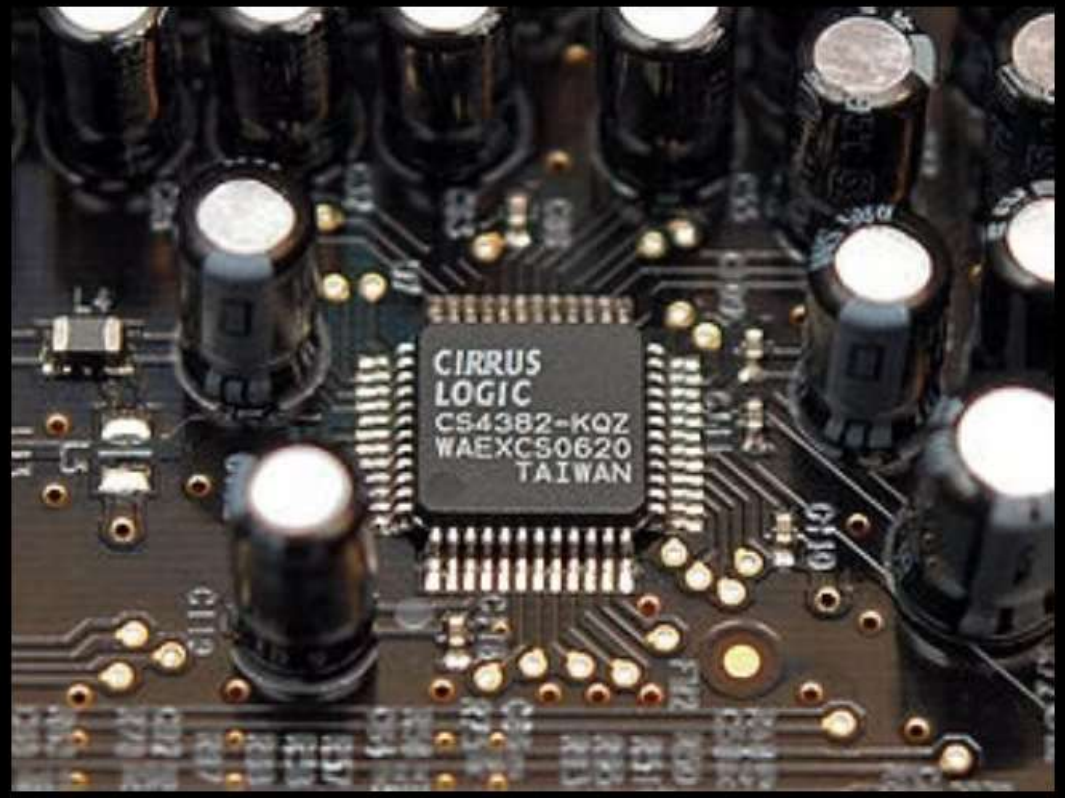
1. Processor
2. DAC(Digital-to-Analog converter)
3. Amplifier
4. Storage device

MICRO PROCESSOR USED IN MP3 PLAYER

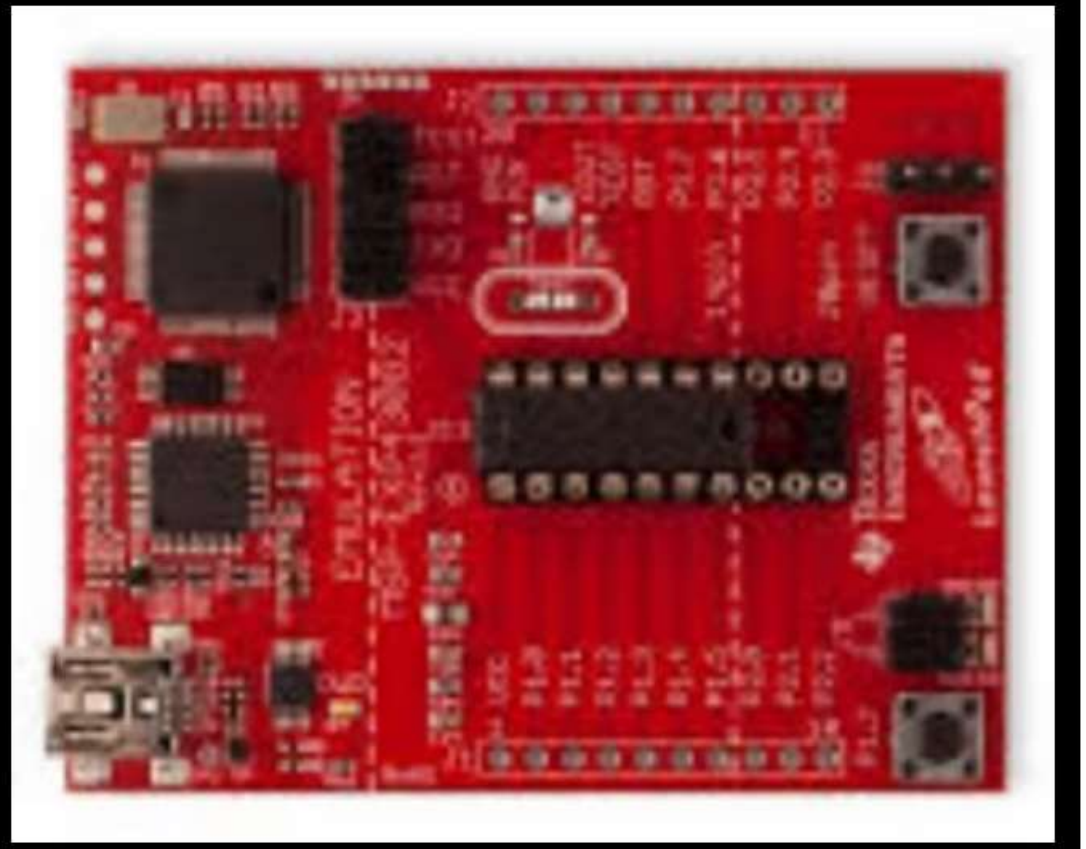
1. ARM Cortex M-series
2. Atmel AVR series

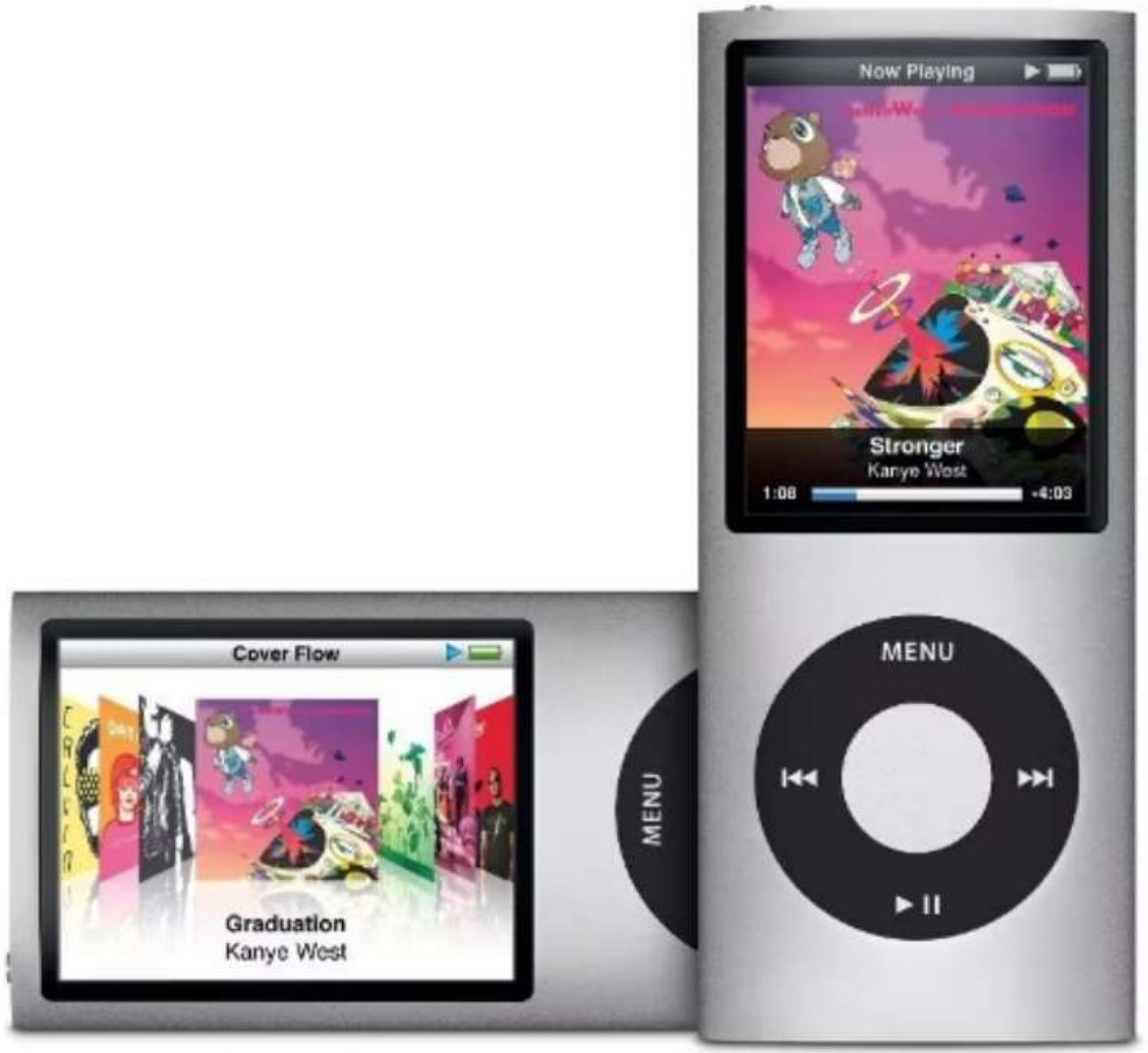



2:40
4:00


CIRRUS
LOGIC
CS4382-KQZ
WAEXCS0620
TAIWAN

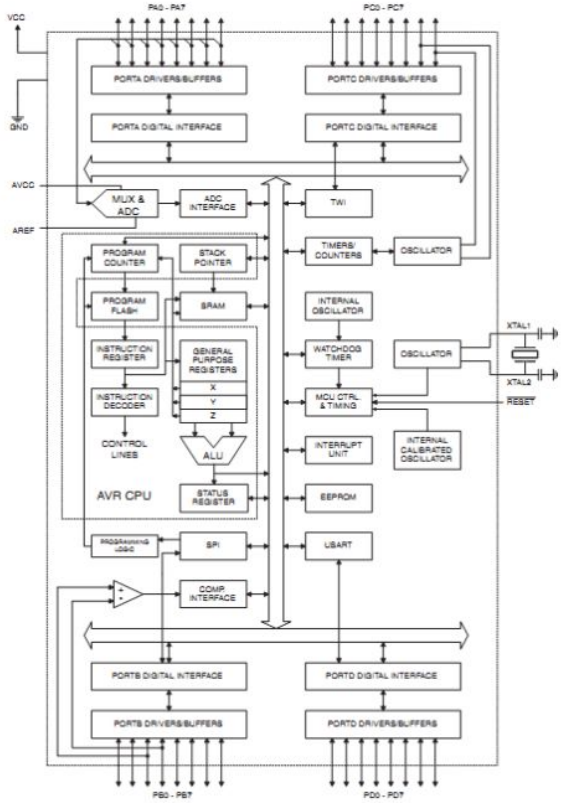
MSP430 and TMS320 Series



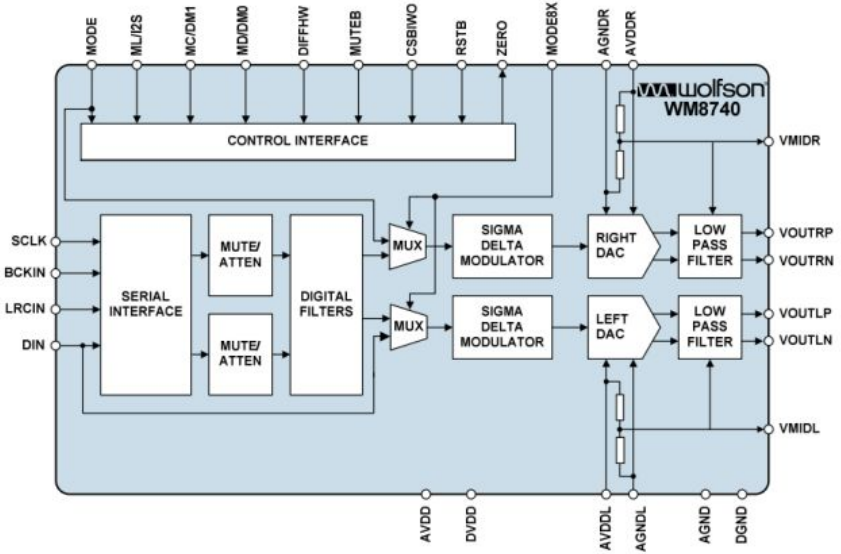

Now Playing
Stronger
Kanye West
1:08
-4:03
MENU
Cover Flow
Graduation
Kanye West
MENU

SYSTEM DESIGN

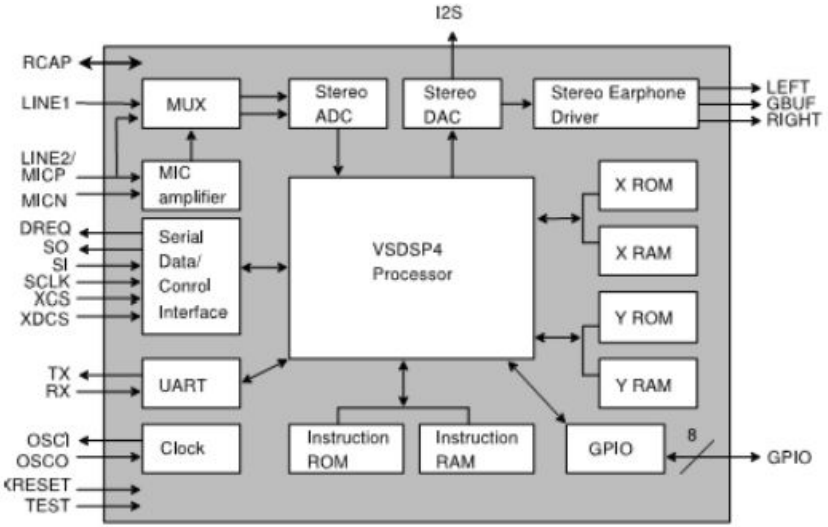
MP3 Player



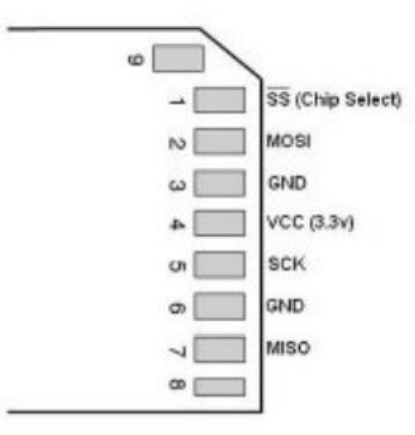
1) Main microcontroller (Atmel ATMEGA32A)



2) Stereo Audio DAC (Wolfson WM8740)



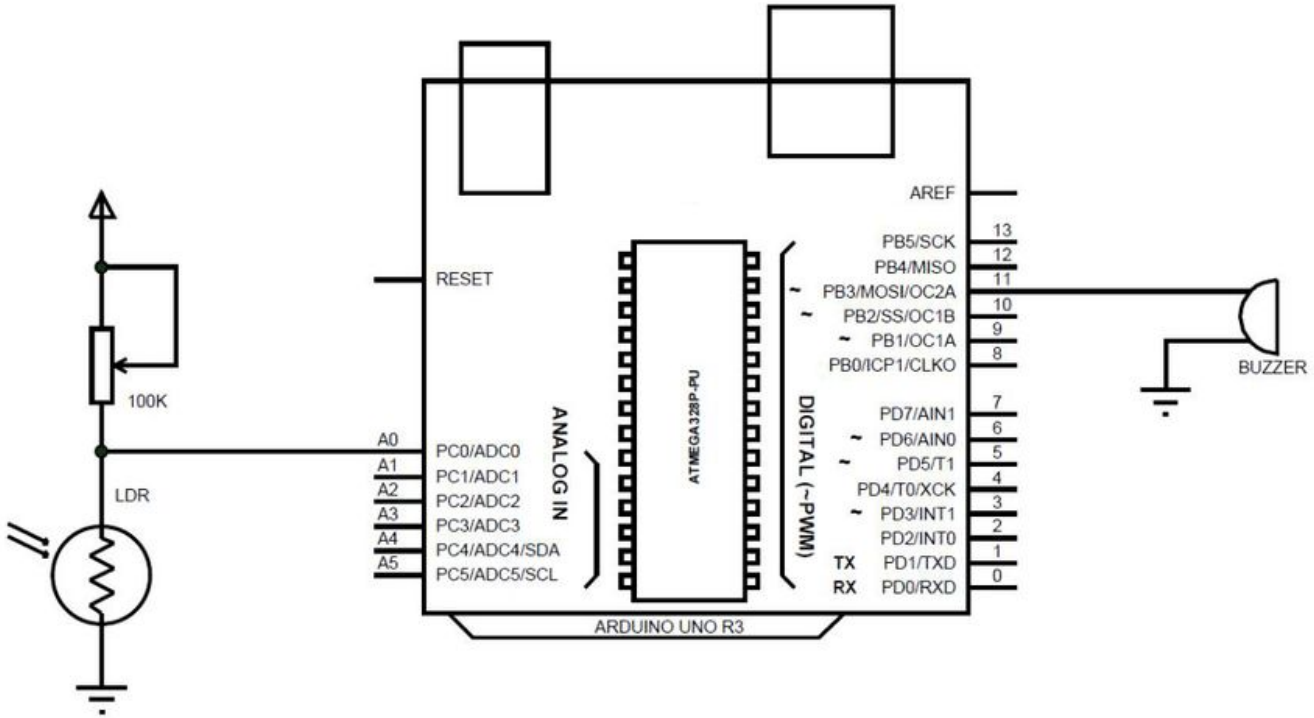
3) Audio Decoder Chip (VLSI VS1053)



4) SD Card/FAT File System

5) Real-Time Operating System (FreeRTOS)

3-D Printer

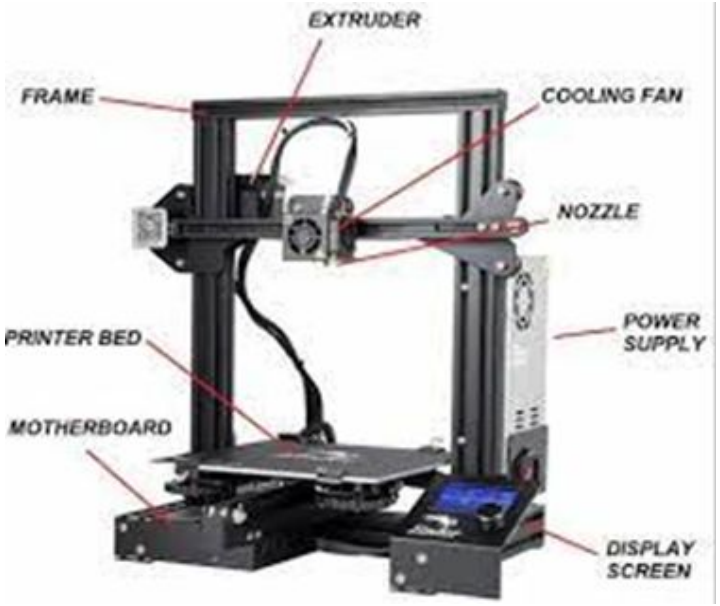


Circuit Diagram Of Arduino Uno R36

CONCLUSION

We considered the two main areas of target here

Broadly classified into two devices

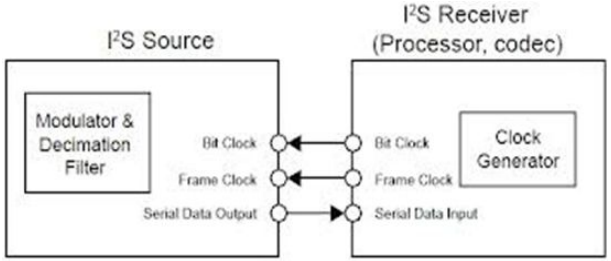


How does a microcontroller helps in playing sound

1) Playing audio using an external sound playback device

All the record and play functions are controlled by the microcontroller i/o ports

2)Playing audio from an inbuilt microcontroller with I2S system



I2S=Inter IC Sound , transfers audio data between chips

Limitation

The main issue is that the Microcontroller's memory utilisation required that the number of timers used be constrained. Increasing the number of timers requires more memory can solve this drawback

3D PRINTER

Made up of two major components

-Automated 3-axis stage controlled by an open source microcontroller

-low cost commercial gas metal arc welder

What is an open source microcontroller

-source code can be modified by enthusiasts

-based around atmel microcontrollers

-uses ide compatible with windows ,mac,linux os

-hardware relatively inexpensive

Limitation

The device is only intended to be used on desktop computers, and it would be better suited to be installed in a garage or shop with sufficient fire protection and ventilation.

Further development of this technology will be extensive in order to prepare it for general use. There are three primary tasks here:

1) electromechanical

2)printer control

3)materials science

Future of microcontrollers

With a thousands of already existing applications in the areas of work,education and healthcare , now microcontrollers will be aiming to transform our life by IOT(Internet of Things).

References

- G. C. Anzalone, C. Zhang, B. Wijnen, P. G. Sanders and J. M. Pearce, "A Low-Cost Open-Source Metal 3-D Printer," in IEEE Access, vol. 1, pp. 803-810, 2013, doi: 10.1109/ACCESS.2013.2293018.

- E. Macdonald et al., "3D Printing for the Rapid Prototyping of Structural Electronics," in IEEE Access, vol. 2, pp. 234-242, Dec. 2014, doi: 10.1109/ACCESS.2014.2311810.

- Basic, Fetah, Sam Roundy, and Mike Ballard. "Mega MP3 Player." (2004).

- A. Schmidt and D. Bial, "Phones and MP3 Players as the Core Component in Future Appliances," in IEEE Pervasive Computing, vol. 10, no. 2, pp. 8-11, April-June 2011, doi: 10.1109/MPRV.2011.31.

- Grazioli, C., Faura, G., Dossi, N., Toniolo, R., Abate, M., Terzi, F., & Bontempelli, G. (2020). 3D printed portable instruments based on affordable electronics, smartphones and open-source microcontrollers suitable for monitoring food quality. *Microchemical Journal*, *159*, 105584. https://doi.org/10.1016/j.microc.2020.105584

- https://digitalcommons.calpoly.edu

Mp3 Player

21BCE0174- Rashmi MT

21BCE0115- Ria Mary Suresh

21BCE0166- Aastha Gupta

21BCT0090- Namish Gupta

21BCE3372- P.Navyanth Reddy

CODE:

```
#include "config.c"

#define oneseg     0x01
#define twoseg     0x02
#define threeseg   0x04
#define fourseg    0x08
#define fiveseg    0x10
#define sixseg     0x20

/*
 *
 */

int change = 0;                              //Default Display(RADIO)
char channels[3][6]={{0x00,0x70,0x77,0x5e,0x06,0xbf},      //Display for RADIO
                     {0x00,0x00,0x00,0x77,0x3e,0xf6},      //Display for AUX
                     {0x00,0x00,0x7c,0x38,0x3e,0xf9}};     //Display for BLUETOOTH
```

```
int temp,tempo;
unsigned long int a,b,c,num;

void interrupt auxx(void);

void main(void){

   OSCCON=0x77;                          //Setting internal oscillator frequency

   TRISBbits.RB0=1;
   TRISBbits.RB1=1;
   TRISBbits.RB2=1;
   TRISBbits.RB7=0;
   TRISC=0x00;
   TRISD=0x00;

   INTCONbits.GIE = 1;
   INTCONbits.PEIE = 1;      //Enable peripheral interrupt
   INTCON2bits.NOT_RBPU = 0;   //Enable pull-up resistor
```

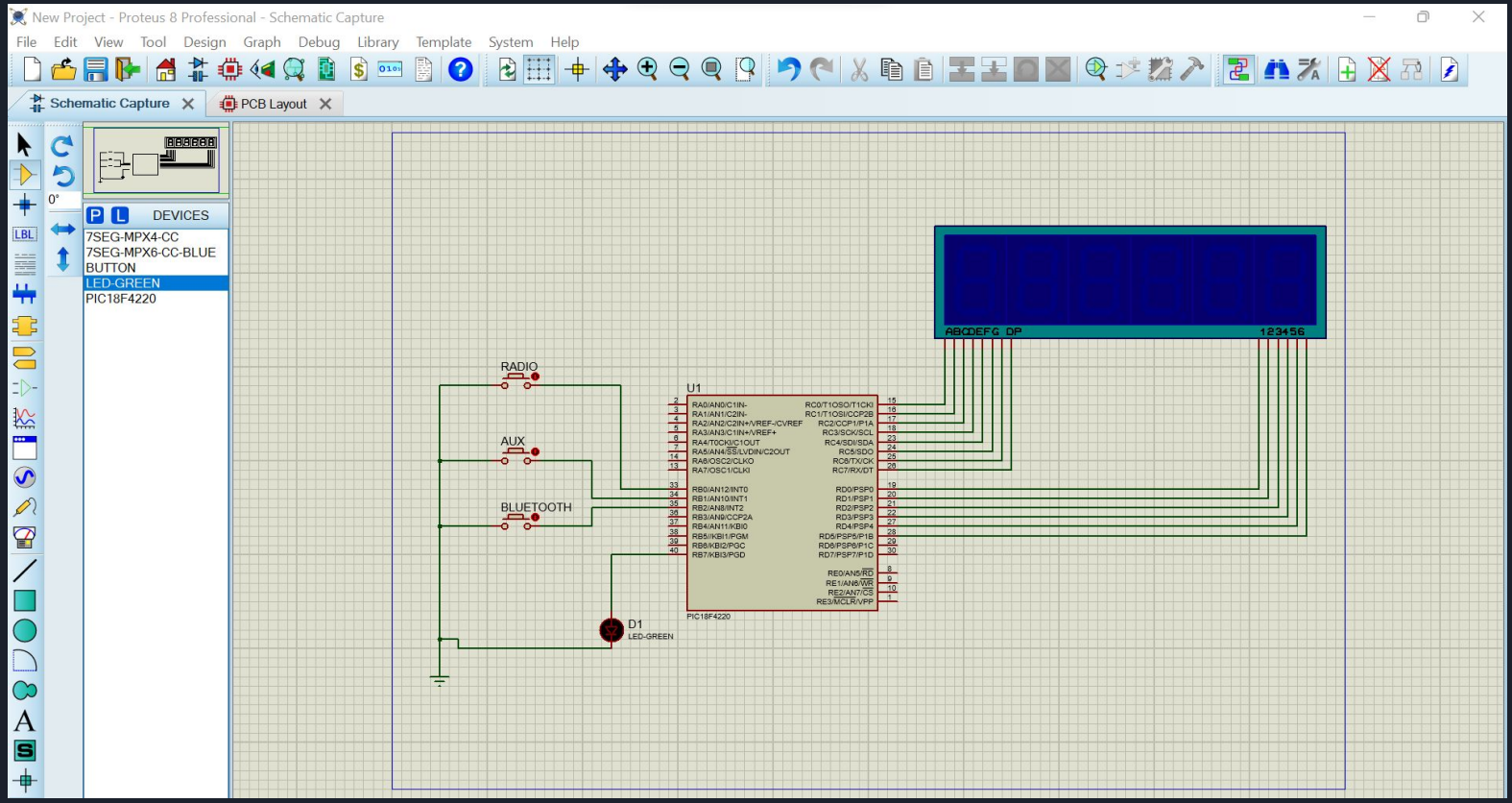
```
//Register configuration for INT0
   INTCONbits.INT0IE = 1;     //Enable interrupt
   INTCONbits.INT0IF = 0;     //Clear flag interrupt
   INTCON2bits.INTEDG0 = 0;   //Interrupt occur at rising edge

   //Register configuration for INT1
   INTCON3bits.INT1IE = 1;    //Enable interrupt
   INTCON3bits.INT1IF = 0;    //Clear flag interrupt
   INTCON2bits.INTEDG1 = 0;   //Interrupt occur at rising edge

   //Register configuration for INT2
   INTCON3bits.INT2IE = 1;    //Enable interrupt
   INTCON3bits.INT2IF = 0;    //Clear flag interrupt
   INTCON2bits.INTEDG2 = 0;   //Interrupt occur at rising edge
   LATBbits.LB7=1;
   while(1){
     for(a = 123456; a == 123456;){
        num = a;
        temp = num / 100000;
        num = num % 100000;
        LATD = ~oneseg;
        tempo = temp-1;
        LATC = channels[change][tempo];
        __delay_ms(3);
```

```
temp = num / 10000;
        num = num % 10000;
        LATD = ~twoseg;
        tempo = temp-1;
        LATC = channels[change][tempo];
        __delay_ms(3);

        temp = num / 1000;
        num = num % 1000;
        LATD = ~threeseg;
        tempo = temp-1;
        LATC = channels[change][tempo];
        __delay_ms(3);

        temp = num / 100;
        num = num % 100;
        LATD = ~fourseg;
        tempo = temp-1;
        LATC = channels[change][tempo];
        __delay_ms(3);
```

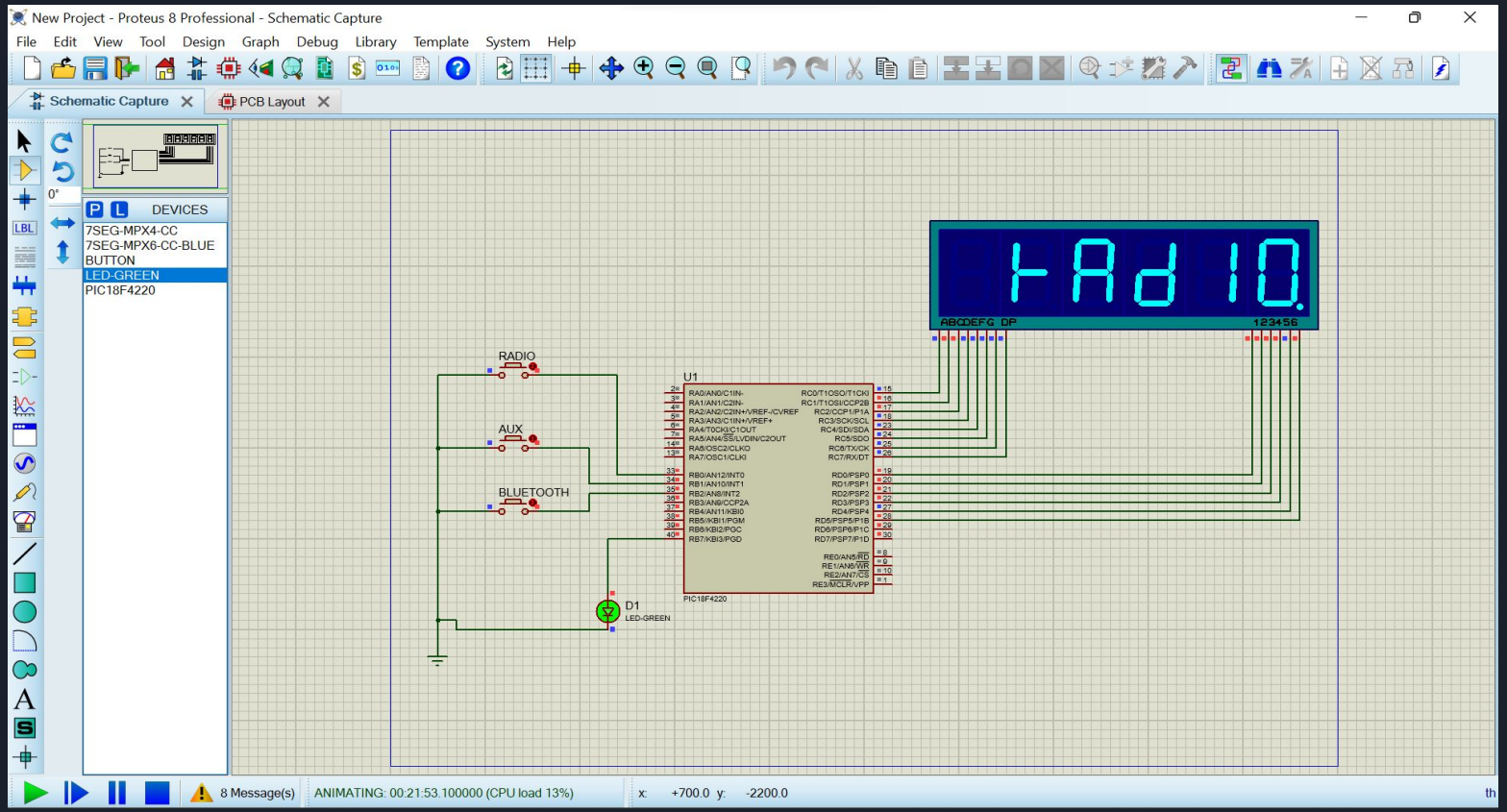
```
temp = num / 10;
        num = num % 10;
        LATD = ~fiveseg;
        tempo = temp-1;
        LATC = channels[change][tempo];
        __delay_ms(3);

        temp = num % 10;
        LATD = ~sixseg;
        tempo = temp-1;
        LATC = channels[change][tempo];
        __delay_ms(3);
      }

   }
}

void interrupt auxx(void){
   if(INTCONbits.INT0IF==1){
      change=0;
      INTCONbits.INT0IF=0;
   }
```

```
if(INTCON3bits.INT1IF == 1){
      change=1;
      INTCON3bits.INT1IF=0;
    }
    if(INTCON3bits.INT2IF == 1){
      change=2;
      INTCON3bits.INT2IF=0;
    }
}
```

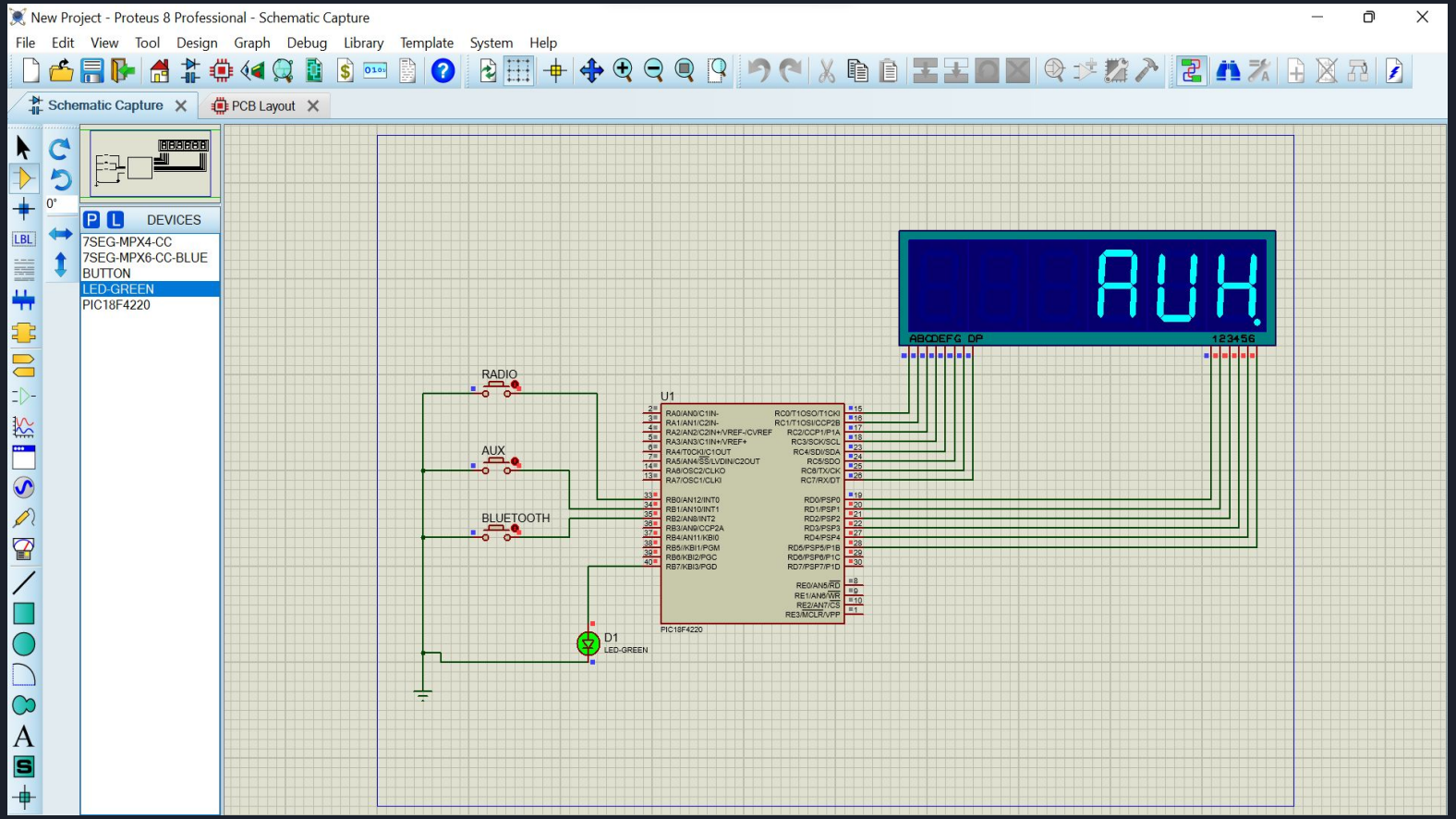
Simulation before running code:



Proteus Simulation after running code:



Simulation when AUX button is pressed:



Simulation when bluetooth button is pressed: