

در این پروژه سعی در فراهم کردن شرایطی را داریم که چندین کلربر توانایی رای دادن داشته باشند و به صورت ۱۰۰ قطعی هیچ رای از دست نرود. برای این کار چند نکته زیر را در نظر بگیرید:

۱. برای این کار اولین نکته حائز اهمیت این است که رای ها بلافاصله پس از داده شدن دانه دانه برای برد مرکزی فرستاده میشوند و در آنجا ذخیره خواهند شد.
۲. هیچ اطلاعاتی بر روی حافظه های برد های slave ذخیره نخواهد شد. برای بالا بردن انعطاف برنامه بنده تمامی اطلاعات کاندید های انتخاباتی را در برد مرکزی ذخیره کرده و در ابتدای اجرای هر کدام از slave ها اطلاعات را دریافت خواهد کرد تا بتوان برای تغییراتی در کاندید های موجود صرفا به کمک تغییر در یک بورد به مورد درخواستی رسید!
۳. برد مرکزی برای کنترل داده های دریافتی و ارسال داده های اولیه الزامی است. بنابراین در صورتی که برد مرکزی با مشکل مواجه شود سایر برد ها شروع به کار نخواهند کرد (زیرا منتظر دریافت اطلاعات اولیه خواهند ماند اما هیچ اطلاعاتی از راه نیرسد)
۴. کلربر قابلیت دیدن اطلاعات اضافی از هر کدام از کاندید ها را خواهد داشت. این اطلاعات عبرت اند از:

- سن

- سال های فعال در حوضه سیاست

- درجه بندی از نظر سیاسی

۵. هنگام رای کلربر برای مدت کوتاهی اطلاعات هر کدام از کلربر ها را میتواند مشاهده کند اما در صورت نیاز میتواند به صفحه اصلی بازگردد و اطلاعات را به صورت مفصل مشاهده کند!

۶. برد مرکزی با رمز پیش فرض ۱۲۳۴ شروع به کار میکند و به مسئول نمایش داده میشود. مسئول میتواند این رمز را تغییر دهد (حداقل ۴ رقم)

۷. از دیگر گزینه های مفید این پروژه امکان مشاهده تمامی رای ها تا کنون و امکان reset کردن تمامی رای ها صرفا برای مسئول رای گیری است.

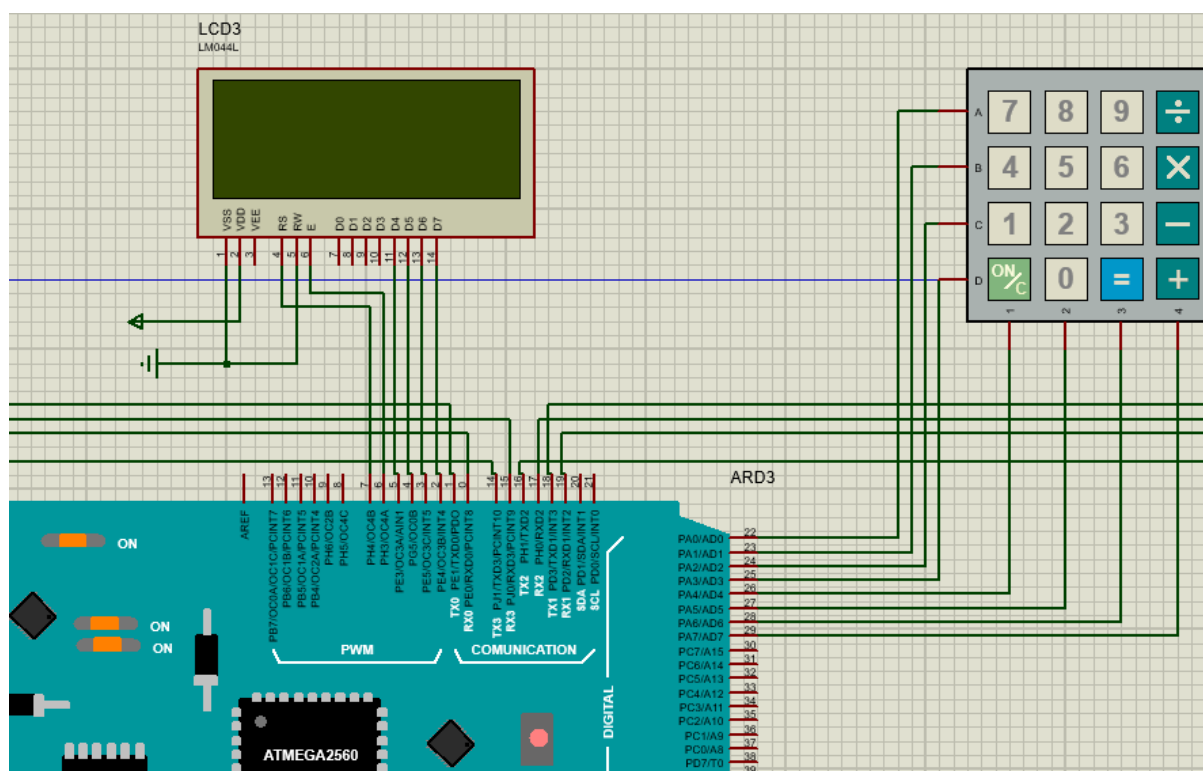
۸. بنده برای راحتی کدی برای بازگردانی به حالت کلرخانه برد مرکزی کنار فایل hex اصلی قرار دادم که میتوانید برای برگرداندن به تنظیمات کلرخانه از آن استفاده کنید.

۹. در این پروژه به دلیل شرایط موجود و تلاش بنده برای از دست رفتن هیچ منظوره داده ها نیاز بود تا هر دو master و slave ها بتوانند با میل خود اطلاعات رد و بدل کنند. بنابراین قادر به پیاده سازی به روش های master – slave مانند I2C و SPI نبودم. بنابراین از پورت های سریال استفاده نمودم!

۱۰. به دلیل مشکل به وجود آمده برای بنده در اواسط ترم در ارتباطات I2C با حافظه های EEPROM بنده با اجاره استاد خود (که از این مشکل در تابع endTransmission به دلیل os یا سخت افزار بنده مطلع بودند) مجبور به یادگیری لایبری EEPROM و استفاده از حافظه داخلی برد مرکزی شدم.

در ادامه به توضیح برخی بخش های مدار ها و بخشهای مهم کد میپردازم:

(۱) بخش مدار :



نمونه ای از مدار ها است. برای هر مدار یک صفحه lcd و یک صفحه کلید برای تعامل با کاربر قرار داده شده است. صفحه lcd را ۲۰ * ۴ انتخاب کردم تا بتوان اطلاعات بیشتری را بتوان نشان داد. در تمامی مراحل از کلید * میتوان برای بازگشت استفاده کرد.

در بخش slave ها میتوان با انتخاب کلید + رای داد یا با کلید / دستورات بیشتر را مشاهده نمود

در بخش master پس از وارد کردن رمز میتوان کارهای ذکر شده در بالا را انجام داد

در قسمت هایی که باید اطلاعاتی وارد شوند از کلید c میتوان برای پاک کردن داده های روی lcd استفاده نمود.

۲) بخش های مهم کد :

```
Serial.begin(9600);
for(int i=0 ; i<4 ; i++)
{
    for(int j=0 ; j<4 ; j++)
    {
        while(Serial.available() == 0);
        candidates[i][j] = Serial.readStringUntil('\n');
    }
}
```

slave

```
Serial.begin (9600);
Serial1.begin(9600);
Serial2.begin(9600);
Serial3.begin(9600);
delay(100);
```

master

```
for(int i=0 ; i<4 ; i++)
{
    for(int j=0 ; j<4 ; j++)
    {
        Serial.println (candidates[i][j]);
        Serial1.println(candidates[i][j]);
        Serial2.println(candidates[i][j]);
        Serial3.println(candidates[i][j]);
        delay(5);
    }
}
```

این بخش کد برای ارتباط اطلاعات اولیه مورد نیاز هر slave که در بخش اول به آن اشاره کردم میباشد.

تمامی اطلاعات در آرایه candidates قرار دارد و در یک حلقه تمامی این اطلاعات فرستاده خواهد شد!

```

byte vote = key - 49;
Serial.write(vote);
lcd.clear();
lcd.setCursor(0 , 1);
lcd.print("  thanks for your");
lcd.setCursor(0 , 2);
lcd.print("          VOTE");
digitalWrite(buzzer_pin , HIGH);
delay(200);
digitalWrite(buzzer_pin , LOW);

```

slave

```

,
void serialEvent()
{
    while(Serial.available())
    {
        value = Serial.read();
        voteNums[value] ++ ;
        EEPROM.write(8+value , voteNums[value]);
    }
}
void serialEvent1()
{
    while(Serial1.available())
    {
        value = Serial1.read();
        voteNums[value] ++ ;
        EEPROM.write(8+value , voteNums[value]);
    }
}
void serialEvent2()
{
    while(Serial2.available())
    {
        value = Serial2.read();
        voteNums[value] ++ ;
        EEPROM.write(8+value , voteNums[value]);
    }
}

```

master

هنگام رای دادن یکی از کاربران در slave رای او به master فرستاده میشود و در EEPROM داخلی او ذخیره میگردد. عدد ۴۹ کم شده از رای به این دلیل است که اعداد بین ۱ تا ۴ و به صورت ASCII هستند پس از آنها ۴۸ واحد کم کرده تا به عدد ۱ تا ۴ برسیم و ۱ واحد دیگر نیز کم میکنیم تا به عددی بین ۰ تا ۳ برسیم (آرایه ما ۴ تایی است)

لازم به ذکر است بخش دریافت اطلاعات به صورت Interrupt پیاده سازی شده است تا به هیچ وجه اولاً رای از دست نرود و ثانياً سربار polling از دوش master برداشته شود تا بتواند سریع تر به رای ها رسیدگی کند!