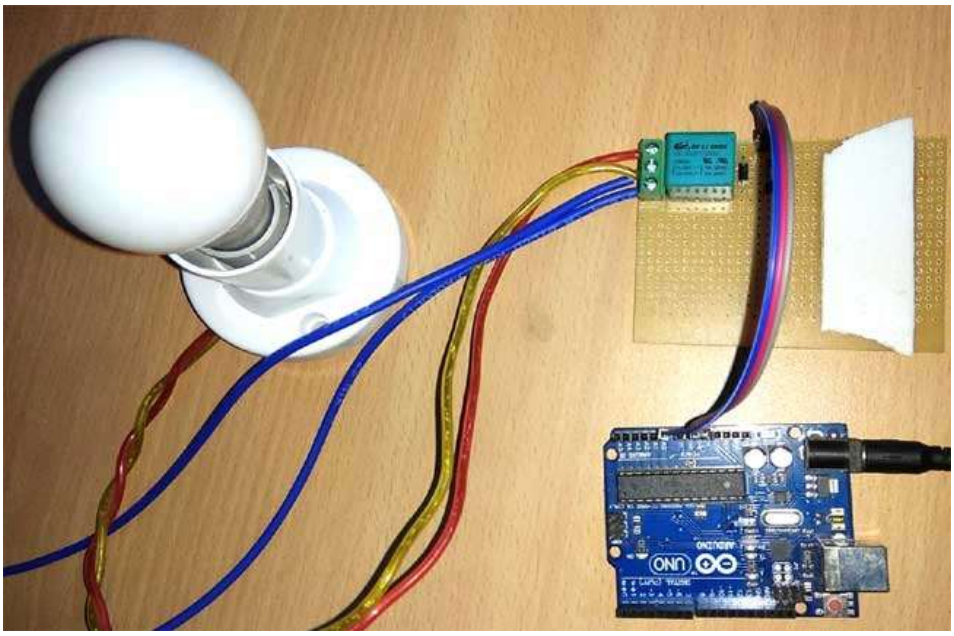
(မှီငြမ်း / **Reference : https://circuitdigest.com/microcontroller-projects/arduino-relay-control** မှ ဘာသာပြန်ဆိုပါသည်။)

Arduino Relay Control Tutorial

အာဒွီနို နှင့် ရီလေးကိုထိန်းချုပ်ခြင်း



LED မီးသီးကို အာဒွီနို နှင့် အမျိုးမျိုး လင်းစေခြင်းကို စတင်လေ့လာသူ အများစု၏ ပထမဆုံး သင်ခန်းစာ အနေဖြင့်လေ့လာကြသည်။ ယခု သင်ခန်းစာ သည် ရီလေး (Relay) ကိုအသုံးပြုပြီး AC မီးသီးကို LED အစား အသုံးပြုနိုင်ရန်ဖြစ်သည်။

ကျွန်ုပ် တို့၏ embedded circuit တွင် AC သုံးကိရိယာ များနှင့်တွဲဖက်အသုံးပြုလိုသောအခါ ရီလေး ကိုအသုံးပြုသည်။ ယခု သင်ခန်းစာသည် အာဒူရီနို နှင့် ရီလေး ကို တွဲဖက်အသုံးပြုရန်နည်းလမ်းကို ဖော် ပြထားသည်။

လိုအပ်သောပစ္စည်းများ

၁။ Arduino Uno

၂။ ၅ ဗို့ သို့မဟုတ် ၆ ဗို့ ရီလေး

၃။ LED မီးသီး

၄။ BC 547 transistor

၅။ 1 K resistor

၆။ Bread board သို့မဟုတ် PCB

၇။ Jumper wire

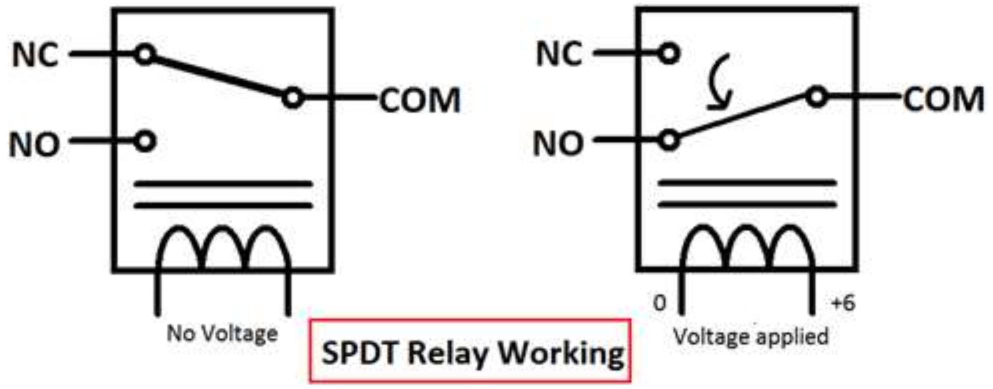
၈။ Power supply

၉။ 1N4007 Diode

၁၀။ Screw terminal or terminal block

<u>ရီလေး (**Relay**)</u>

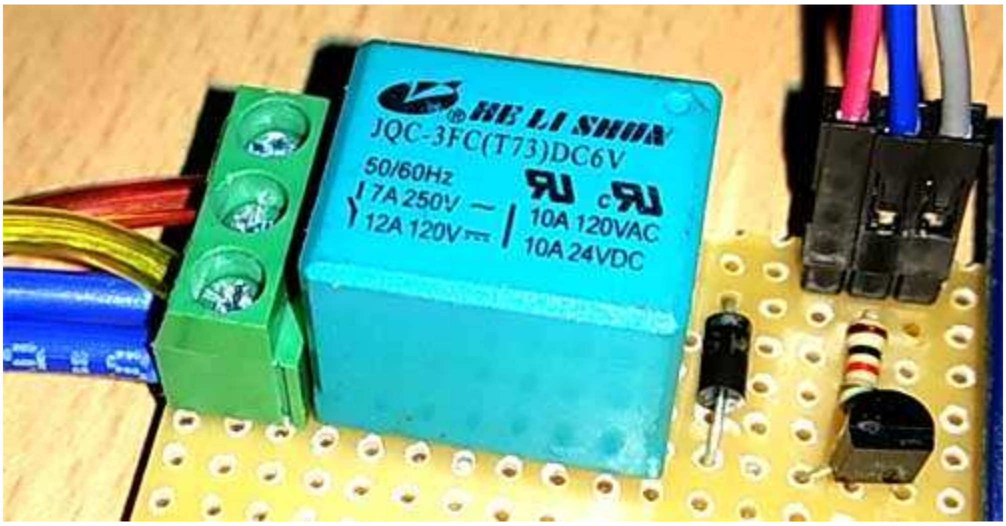
ရီလေးဆိုသည်မှာ လျှပ်စီးအနည်းငယ်ဖြင့် ထိန်းချုပ် နိုင်သော လျှပ်စစ်သံလိုက် သုံးခလုပ် တစ်ခုဖြစ်သည်။ ၄င်း ကို ပို၍ ကြီးသော လျှပ်စီး ပတ်လမ်းများကို အဖွင့် အပိတ်ပြုလုပ်ရန် လည်း အသုံးချနိုင်သည်။ ရီလေးသည် AC လျှပ်စစ် သုံးကိရိယာ နှင့် ပတ်လမ်းကို DC လျှပ်စီး အနည်းငယ်ဖြင့် ထိန်းချုပ်နိုင်သည်ကို ပြသသောအကောင်းဆုံးဥပမာ ဖြစ်သည်။ အများဆုံးအသုံးပြုလေ့ရှိသော ရီလေးအမျိုးအစားမှာ Single Pole Double Throw (SPDT) relay အမျိုးအစားဖြစ်သည်။



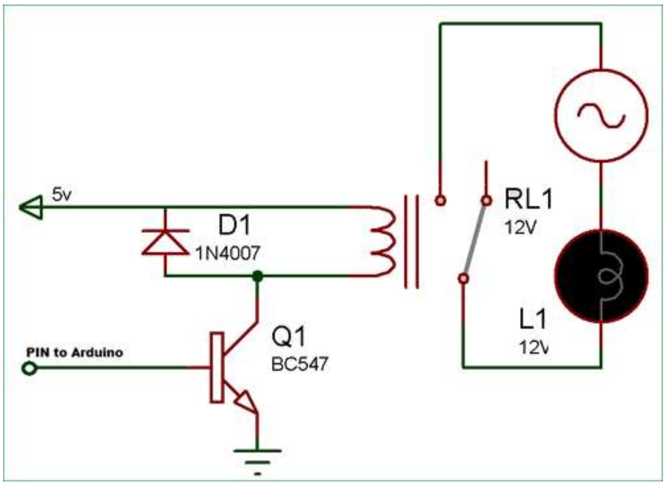
အကယ်၍ ကွိုင်ကြိုးစ သို့မည်သည့် ဗို့အားမျှ ပေးသွင်းမထားလျှင် COM (common) သည် NC (Normally closed contract) နှင့်သာ ချိတ်ဆက်နေမည်။ အကယ်၍ ဗို့အား တစ်ခုကို ကွိုင်ကြိုးစ သို့ ပေးသွင်းလိုက်လျှင် လျှပ်စစ်သံလိုက်လှိုင်းဖြစ်ပေါ်လာပြီး အာမေချာ (Armature (lever connected to spring) ကို ဆွဲငင် မှုဖြစ်စေသောကြောင့် COM နှင့် NO (normally open contact) ကိုချိတ်ဆက်စေသည်။ ထိုကဲ့သို့ချိတ်ဆက်မှုကြောင့် ပို၍ကြီးသော လျှပ်စီး စီးဆင်းမှုကို ဖြစ်ပေါ်စေသည်။ ရီလေး ကိုအသုံးချမည့် နေရာပေါ်မူတည်၍ ခံနိုင်ဝန် နှင့် ဗို့အား ကွာခြားမှုရှိသည်။ ယခုသင်ခန်းစာ တွင် ၆ ဗို့ ရီလေးကို အသုံးပြုမည်ဖြစ်ပြီး ခံနိုင်ဝန် ၇ အမ်ပီယာ/ ၂၅၀ အေစီ ဗို့ဖြစ်သည်။

ရီလေး ကိုမောင်းနှင်ရာတွင် ထရန်စစ်စတာ၊ ဒိုင်အုပ် နှင့် ရီစစ်စတာ တို့ပါဝင်သော driver circuit ကို အသုံးပြုကြသည်။ ထရန်စစ်စတာ သည် လျှပ်စစ် ကိုချဲ့ထွင် ပေးခြင်းဖြင့် ကွိုင်အတွင်းဖြတ်စီး သောလျှပ်စီးကို အပြည့်အဝ အသုံးပြုနိုင်စေသည်။ ရီစစ်စတာ သည် ထရန်စစ်စတာ အလုပ်လုပ်

နိုင်စေရန်အတွက် သင့်လျော်သော စွမ်းအားပေးရန်ဖြစ်သည်။ ဒိုင်အုပ် သည် ထရန်စစ်စတာ switch off တွင်ဖြစ်ပေါ်သော ပြောင်းပြန်လျှပ်စီးကို ကာကွယ်ရန်ဖြစ်သည်။ ကွိုင်များသည် ရုတ်တရက် ပိတ်လိုက်သောအခါ တူညီသော ဆန့်ကျင်ဘက် လျှပ်စစ်သံလိုက်စွမ်းအား ကိုထုတ်လုပ်ပေးသည်။ အဆိုပါ လျှပ်စစ်သံလိုက်စွမ်းအား သည် ပြောင်းပြန်လျှပ်စီးအဖြစ် လျှပ်စီးပတ်လမ်းတွင်းရှိ ပစ္စည်းများကို ပျက်စီးစေနိုင်သည်။ ထိုသို့သောပျက်စီးမှုကို ကာကွယ်နိုင်ရန် ဒိုင်အုပ်ကို အသုံးပြုသည်။
ရီလေးမိုဂျူး (Relay Module) သည် အထက်ပါ ရီလေးပတ်လမ်းကို အသင့်တည်ဆောက်ထားပြီး ဖြစ်သည်။

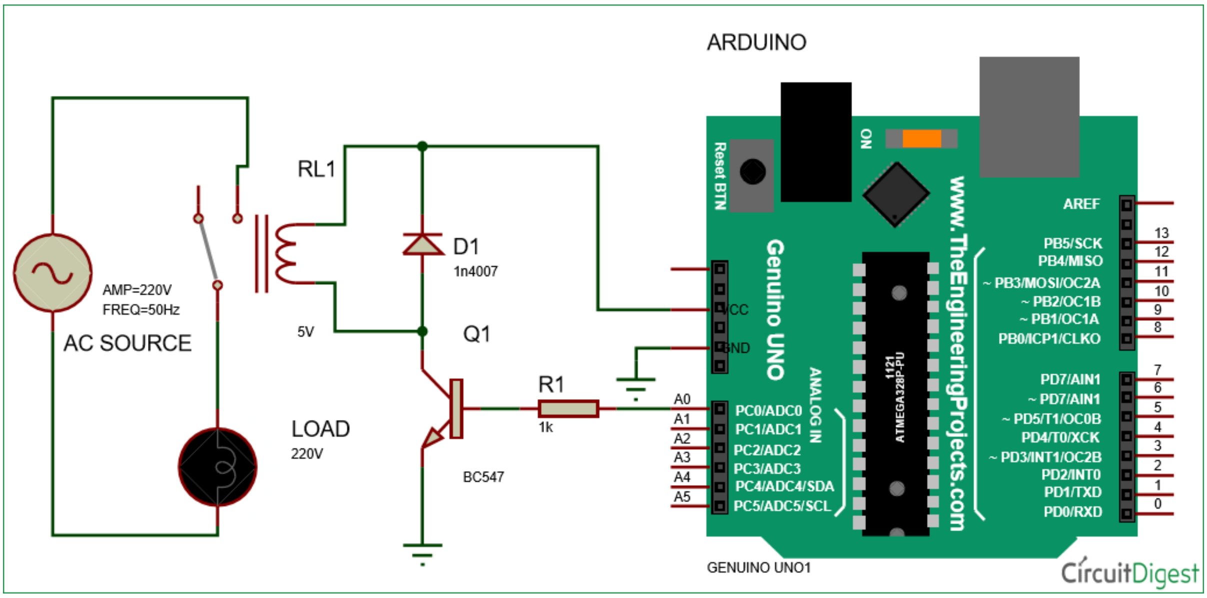


အာဒွီနို ဖြင့် ရီလေး တစ်လုံးကို အသုံးပြုရန် အာဒွီနို နှင့် ရီလေး ဆက်သွယ်ထားသောပင်ငုတ် (Pin, ယခုသင်ခန်းစာ တွင် A0) ကို High ဖြစ်စေမည်။
အောက်ပါပုံသည် ရီလေးမိုဂျူး တစ်ခုကို ကိုယ်တိုင် တည်ဆောက်ရာတွင်အသုံးပြုနိုင်သည့်ပတ်လမ်းဖြစ်သည်။



Relay Driver Circuit

ပတ်လမ်းနှင့်အလုပ်လုပ်ပုံ



ဤ အာဒွီနို နှင့် ရီလေး ပတ်လမ်းတွင် အာဒွီနို သည် ရီလေး ကို ထရန်စစ်စတာ မှတစ်ဆင့်ထိန်းချုပ်သည်။ ထရန်စစ်စတာ ၏ Base ကို အာဒူရီနို ၏ ပင်ငုတ် A0 နှင့် 1K ရီစစ်စတာဖြင့်ဆက်သွယ်ထားသည်။ AC မီးသီးသည် ရီလေး အဖွင့် အပိတ်ကို သရုပ်ပြရန်ဖြစ်ပြီး ၁၂ ဗို့ adapter သည် ပတ်လမ်းကို ဓါတ်အားပေးရန်ဖြစ်သည်။

အလုပ်လုပ်ပုံမှာ ရီလေး မိုဂျူး အသက်ဝင်စေရန် အာဒွီနို ၏ ပင်ငုတ် A0 ကို High ဖြစ်စေခြင်းဖြစ်သည်။ ပင်ငုတ် A0 High ဖြစ် လျှင် ရီလေး ချိတ်ဆက်ပြီး ပင်ငုတ် A0 Low ဖြစ်လျှင် ရီလေး၏

ချိတ်ဆက်မှု ပြတ်တောက်သွားမည်။ High ဖြစ်ချိန်တွင် လျှပ်စစ် မီးသီး လင်းလာမည်ဖြစ်ပြီး၊ Low ဖြစ်ချိတ်တွင် လျှပ်စစ်မီးသီး မှိတ်သွားမည်။

ကျွန်ုပ် တို့သည် အာဒွီနို နှင့် ရီလေးဆက်သွယ်သော ပင်ငုတ် A0 ၏ High နှင့် Low ကို ၁ စက္ကန့် delay ဖြင့် သတ်မှတ်ပါသည်။

```
void loop()

{

   digitalWrite(relay, HIGH);

   delay(interval);

   digitalWrite(relay, LOW);
```

Code အပြည့်အစုံမှာ အောက်ပါအတိုင်း ဖြစ်သည်။

```
// Arduino Relay Control Code

#define relay A0
#define interval 1000
void setup() {
  pinMode(relay, OUTPUT);
}
void loop()
{
   digitalWrite(relay, HIGH);
   delay(interval);
   digitalWrite(relay, LOW);
   delay(interval);
}
```

အောက်ပါ Link တွင် သင်ခန်းစာ ၏ ဗွီဒီယို ကိုကြည့်ရှု နိုင်ပါသည်။

https://youtu.be/fA0VkfPcREU