

Here, the control circuits that the output pins are connected to is not treated, I know so well what I will do with them. That side is no problem with me. The area I want a paid developer is input side. I will post a demo circuit representing the 9 Lines of the three Sources with equivalent DC voltages, LOW Level, Accepted level and High level with Proteus simulation files and Hex. That is animated representation of this whole project though I used serial display instead of LCD in the demo.

The system will be monitoring the status of 3 Lines of the three sources. The national/city grid is the top priority of the three sources. No matter which source is on whenever the city grid is restored.

All the delays I used within the loop are non-blocking

SOURCE1 IS OK.

- The system will monitor it for **15 seconds** if all is well, that is, if the city grid is not below a pre-defined low level or higher than a pre-defined high level within that 15 seconds, the system will turn on the **LED_OUTPUT-S1** while **LED-OUTPUT-S2** and **LED-OUTPUT-S3** will become **LOW** whichever that is on. Then after another **15 seconds** is elapsed, the **LED-RELAY-S1** will be **HIGH**.

In this mode;

int8 *source1*=1 (*system=source1 mode*)

int8 *source2*=0

int8 *source3*=0

(LCD displays SOURCE1 parameters)

- If all is not okay for consecutive 15 seconds, the system will continue to ignore Source1 while Source2 or Source3 whichever that is on, continue to supply.

In this mode;

int8 *source1*=0

int8 *source2* or *source3*=1 whichever that is on

int8 *system=source2 mode* or *source3 mode* whichever that is on.

SOURCE2 IS OK – (SOURCE1 and SOURCE3 are absent or within unaccepted range)

- The system will wait for 15 seconds and after if the source2 is approved, **LED-OUTPUT-S2** will go HIGH.
- If **MODE-SELECT-SW** is **HIGH** (source2 is master to source3), **LED-RELAY-S2** will go **HIGH** after another 15 seconds.
- If the **MODE-SELECT-SW** is **LOW**, **LED-RELAY-S2** will remain **LOW**.
- In this mode;
- **int8** *source1*=0
- **int8** *source2*=1 (*system=source2 mode*)
- **int8** *source3*=0
- (LCD displays SOURCE2 parameters)

WHERE FAST INSTRUCTION TIME IS NEEDED

During all these previous operations, delays caused by RMS calculations does not matter. It only matters on the **SOURCE REVERT**. For example, if the system is in **SOURCE2 MODE**, and slave, if **SOURCE3** comes on, the system will migrate to **SOURCE3** and the **SOURCE2** generator will wait for 15 seconds before the system will send a turning off signal to it. The reason why it will wait for 15 seconds is in case the **SOURCE3** went off by any reason within that 15 seconds wait. For instances within that 15 seconds turn-off wait and you turn off the main breaker the system will immediately **REVERT** to the **SOURCE2** which has not gone off and the delay will be cancelled.. One may see it as unnecessary but I have seen it messed things before.