[System Architecture](#h.h98c2qbs3qjk)

[Hardware Architecture](#h.x534mnp37c8o)

[PCB Architecture](#h.w39lz65cadsn)

[Firmware Architecture](#h.i5bxigg37hib)

# System Architecture



STATION:

**ANDON SYSTEM**



# 

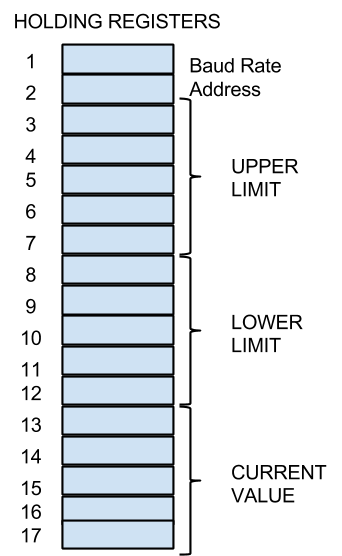
# Specifications

1. Lines : 5
2. Visibility Range : 20 meters
3. Single Sided
4. Data : 4 Digits
5. Indicators :
   1. visual : Lamps for each line
   2. audio : Buzzer - common for all lines
6. Communication Protocol : MODBUS
7. Roof Mounting

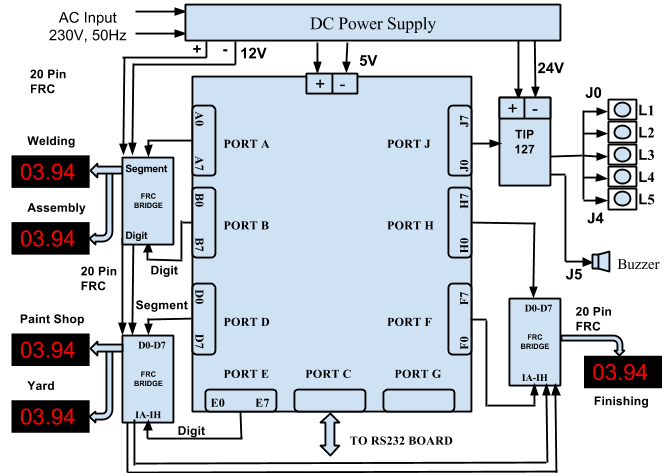
# Hardware Architecture



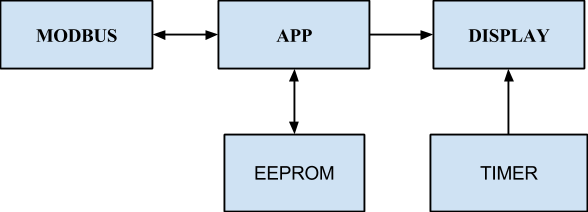
# Modbus Register Map



# PCB Architecture



# Firmware Architecture



# Application Algorithm

appstructure

{

baudRate;

baudrateUpdated; //

address;

addressUpdated;

limitBuffer[MAX\_LINES\*2];

limitUpdated;; // whenever the start address is between 3 -12 we will set this flag

valueBuffer[MAX\_LINES];

valueUpdated[MAX\_LINES]; // to store flag of each lines

}

## a. Holding Register Callback

1. Check for starting address and number of registers not be zero.
2. if baudrate updated store new baudrate and baudrateUpdated flag to true
3. if address updated store new address and set addressUpdated flag to true
4. If starting address is greater than 3 and less than 12, store the content of buffer into app buffer and set limitUpdated flag.
5. If the starting address is greater than 12, store the content of buffer into a app buffer and set corresponding line flag.
6. Repeat step 4 and 5 till Number Register count becomes zero.

## b. App Task

1. if baudrateUpdated is true, update eeprom and reset the flag. Reinitialize Modbus stack
2. if addressUpdated is true, update eeprom and reset the flag. Reinitialize Modbus stack
3. if limitUpdated flag is true, update eeprom for all lines and reset the flag
4. for each line, if valueUpdated is true, update display.
5. update indicators