**Taking good parts of all approaches we have this:**

* Knob is controlling the parameter.
* Sense the parameter first by the microcontroller.
* Ask the user to use the knob to the max position (meaning low resistance as in a pot)
* So in effect find out the resistance range (max and min)

All this with switch (main fan switch) off.

ITS like a boot time check(only once done during installation).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Micro Controller controls the speed of the fan based on two inputs.

**A)Input from Phone (In\_Android)**. **(B) Input from wall Knob (In\_Knob).**

* Among the two input **In\_Knob** will have higher priority (Works either way) only the first time(during **boot time** if in case of collision)
* If no input is detected then the Microcontroller runs the fan at previous speed.
* **In\_Android** changes and **In\_Knob** does not change- Under this condition Microcontroller changes fan speed according to change in **In\_Android**--So is the **In\_Knob** position changed too.(due to the motor movement)
* **In\_Knob** changes and **In\_Android** does not change- Under this condition Microcontroller changes fan speed according to change in **In\_Knob** and the current fan speed is relayed back to the android phone.
* **In\_Android** changes and **In\_Knob** also changes- Under this condition Microcontroller changes fan speed according to change in In\_Knob and ignores **In\_Android**--(this cannot happen as if the android phone is triggering a change the same change will be echoed back at the knob due to motor movement making it the user impossible to change the wall knob position when android phone is being used).

In Summary -

This approach has the mixed advantages of all the approaches with elimination of almost all the disadvantages.

It *achieves*:

* **Range detection and feedback control (so it doesn't flip-out).**
* **Simultaneous sync movement so there is no problem of collision whatsoever.**
* **Less analog circuitry involved to an extent(Yes there is power used but I think it will be meagre since a DC motor uses very less current to rotate(and we don't want high speed either so less current).**

**\*\*\*Using Motor circuit to rotate the knob in either direction :**

**Circuitry involved:**

* **Better to use the AC rectified DC circuit to power up the motor .**
* **Needs a AC - DC smooth rectifier and a controlling setting to vary motor according to Android data.**

**Alternative :**

**Instead of using a motor we can use a RC network(mostly yet to be ascertained) or in the same likes a simple circuit where either C/R is varying depending upon android data.**

**A digital switch is involved which connects the existing knob circuitry to this** RC network**(or some sort of network) only when speed is changed at the PHONE side(In\_Android).**

**\*\*\* This is tentative since we don't know what to do :P**